SILVACO International

ATLAS User's Manual

DEVICE SIMULATION SOFTWARE

Volume I

SILVACO International
4701 Patrick Henry Drive, Bldg. 1
Santa Clara, CA 94054
Telephone (408) 567-1000
FAX: (408) 496-6080

November 1998

ATLAS User's Manual

Copyright 1998
SILVACO International
4701 Patrick Henry Drive, Building 1
Santa Clara, CA 95054

Phone: (408) 567-1000
FAX: (408) 496-6080

Notice

The information contained in this document is subject to change without notice.

SILVACO International MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE.

SILVACO International Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

This document contains proprietary information, which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of **SILVACO** International.

Simulation Standard, TCADDrivenCAD, Virtual Wafer Fab, Analog Alliance, Legacy, ATHENA, ATLAS, FASTATLAS, ODIN, VYPER, CRUSADE, RESILIENCE, DISCOVERY, CELEBRITY, Production Tools, Automation Tools, Interactive Tools, TonyPlot, DeckBuild, DevEdit, Interpreter, ATHENA Interpreter, ATLAS Interpreter, Circuit Optimizer, MaskViews, PSTATS, SSuprem3, SSuprem4, Elite, Optolith, Flash, Silicides, SPDB, CMP, MC Deposit, MC Implant, Process Adaptive Meshing, S-Pisces, Blaze, Device3D, Thermal3D, Interconnect3D, Blaze 3D, Giga3D, MixedMode3D, TFT, Luminous, Giga, MixedMode, ESD, Laser, FastBlaze, FastMixedMode, FastGiga, FastNoise, MOCASIM, UTMOST, UTMOST II, UTMOST III, UTMOST IV, PROMOST, SPAYN, SmartSpice, MixSim, Twister, FastSpice, SmartLib, SDDL, EXACT, CLEVER, STELLAR, HIPEX, Scholar, SIREN, Escort, Starlet, Expert, Savage, Scout, Guardian, and Envoy – are trademarks of **SILVACO** International.

© 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997, 1998 by **SILVACO** International Inc.

Reader Comment Sheet

We welcome your evaluation of this manual. Your comments and suggestions help us to improve our publications. If you have any responses to the questions below, please let us know. Write your observations, complaints, bug reports, suggestions or comments below.

- Is this manual technically accurate?
- Are the concepts and wording easy to understand?
- Is the size of this manual convenient for you?
- Is the manual's arrangement convenient for you?
- Do you consider this manual to be easily readable?
- Please add any additional relevant comments.

Please fax your comments to **SILVACO** International, Attention Technical Publications at (408) 496-6080.

Introduction

Intended Audience

The information presented is based on the assumptions that the reader is (1) familiar with the basic terminology of semiconductor processing and semiconductor device operation, and (2) understands basic operation of the computer hardware and operation system being employed.

Introduction

ATLAS is a modular and extensible framework for one, two and three dimensional semiconductor device simulation. It is implemented using modern software engineering practices that promote reliability, maintainability, and extensibility. Products that use the ATLAS Framework meet the device simulation needs of all semiconductor application areas.

SILVACO International makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranty of fitness for a particular purpose.

SILVACO International shall not be liable for errors contained herein or for incidental or consequential damages in connection with furnishing, performance, or use of this material. This document contains proprietary information protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated into another language without the prior written consent of **SILVACO** International.

Editions are recorded below under History, and are individually listed as Edition 1 through 6. The basic issue of the manual is Edition 1. The date is also noted. A completely revised manual results in a new edition.

History

- Edition 1 - July 1, 1993
- Edition 2 - March 1, 1994
- Edition 3 - June 1, 1994
- Edition 4 - October 30, 1996
- Edition 5 - April 30, 1997
- Edition 6 - November 1998

This page intentionally left blank.

Table of Contents

Chapter 1: Introduction 1-1

Overview of ATLAS 1-1

Organization Of This Manual 1-3

Technical Support 1-3

Features And Capabilities of ATLAS 1-4

A Comprehensive Set of Models: 1-4

Fully Integrated Capabilities: 1-4

Sophisticated Numerical Implementation 1-4

Using ATLAS With Other Silvaco Software 1-5

The Nature Of Physically-Based Simulation 1-6

Chapter 2: Getting Started with ATLAS 2-1

Overview 2-1

ATLAS Inputs and Outputs. 2-1

Modes of Operation 2-2

Interactive Mode With DeckBuild 2-2

Batch Mode With DeckBuild 2-2

No Windows Batch Mode With DeckBuild 2-3

Running ATLAS inside Deckbuild 2-3

Running a given version number of ATLAS 2-3

Starting Parallel ATLAS 2-3

Batch Mode Without DeckBuild 2-3

Accessing The Examples 2-4

The ATLAS Syntax 2-5

Statements and Parameters. 2-5

The Order of ATLAS Commands 2-6

The DeckBuild Command Menu 2-7

Quick Start for PISCES-II Users 2-7

Defining A Structure 2-8

Interface From ATHENA 2-8

Interface From DevEdit 2-9

Using The Command Language To Define A Structure 2-9

Specifying The Initial Mesh 2-9

Specifying Regions And Materials 2-10

Cylindrical Coordinates 2-11

Specifying Electrodes 2-11

Specifying Doping 2-11

Analytical Doping Profiles 2-12

Importing 1D SSUPREM3 Doping Profiles 2-13

Remeshing Using The Command Language 2-13

Regrid On Doping 2-13
Regrid Using Solution Variables 2-14
Specifying 3D Structures 2-15
General Comments Regarding Grids 2-15
Maximum Number Of Nodes 2-16
Defining Material Parameters And Models 2-17
Specifying Contact Characteristics 2-17
Workfunction for Gates or Schottky Contacts 2-17
Setting Current Boundary Conditions 2-18
Defining External Resistors, Capacitors or Inductors 2-18
Floating Contacts 2-18
Shorting two contacts together 2-19
Making an open circuit contact 2-20
Specifying Material Properties 2-20
Setting Parameters 2-20
Heterojunction Materials 2-21
Specifying Interface Properties 2-21
Specifying Physical Models 2-21
Energy Balance Models 2-22
Summary Of Physical Models 2-23
Using The C-Interpreter To Specify Models 2-26
Choosing Numerical Methods 2-27
Numerical Solution Techniques 2-27
Basic Drift Diffusion Calculations 2-27
Drift Diffusion Calculations with Lattice Heating 2-27
Energy Balance Calculations 2-28
Energy Balance Calculations with Lattice Heating 2-28
Setting The Number Of Carriers 2-28
Important Parameters Of the METHOD Statement 2-29
Restrictions on the Choice of METHOD 2-29
Pisces-II Compatibility 2-30
Obtaining Solutions 2-31
DC Solutions 2-31
Sweeping The Bias 2-31
Generating Families Of Curves 2-32
The Importance Of The Initial Guess 2-32
The Initial Solution 2-33
The First And Second Non-Zero Bias Solutions 2-33
The Trap Parameter 2-33
Small-Signal AC Solutions 2-34
Single Frequency AC Solution During A DC Ramp 2-34
Ramped Frequency At A Single Bias 2-34
Transient Solutions 2-35
Advanced Solution Techniques . . . . . -36
Obtaining Solutions Around The Breakdown Voltage 2-36
Using Current Boundary Conditions 2-36
The Compliance Parameter 2-37
The Curvetrace Capability 2-37

Using DeckBuild To Specify SOLVE Statements . . . 2-38
Interpreting The Results . . . 2-38
Run-Time Output . . . 2-38
Log Files . . . 2-40
Units Of Currents In Log files . . . 2-40
Parameter Extraction In DeckBuild . . . 2-40
Functions In TonyPlot . . . 2-41
AC Parameter Extraction . . . 2-42
UTMOST Interface . . . 2-42
Solution Files . . . 2-42
Interpreting Contour Plots . . . 2-43
Customizing Solution Files (OUTPUT Statement) . . . 2-43
Saving Quantities from the Structure at each Bias Point (PROBE statement) . . . 2-43
Re-initializing ATLAS at a Given Bias Point . . . 2-44
Technology Specific Issues in ATLAS . . . 2-44

Chapter 3:
Physics . . . 3-1

Basic Semiconductor Equations . . . 3-1
Overview . . . 3-1
Poisson's Equation . . . 3-1
Carrier Continuity Equations . . . 3-1
The Transport Equations . . . 3-2
The Drift-Diffusion Transport Model . . . 3-2
Energy Balance Transport Model . . . 3-4
Displacement Current Equation . . . 3-4
Basic Theory of Carrier Statistics . . . 3-5
Fermi-Dirac and Boltzmann Statistics . . . 3-5
Effective Density of States . . . 3-5
Intrinsic Carrier Concentration . . . 3-6
The Energy Bandgap . . . 3-7
Bandgap Narrowing . . . 3-7
Space Charge from Incomplete Ionization, Traps and Defects . . . 3-9
Overview 3-9 . . .
Incomplete Ionization of Impurities . . . 3-9
Low Temperature Simulations . . . 3-10
Traps and Defects . . . 3-11
Overview . . . 3-11
Calculation of Trapped Charge in Poisson's Equation . . . 3-12
Trap Implementation into Recombination Models . . . 3-13
Trap-Assisted Tunneling . . . 3-14
Transient Traps . . . 3-16
The Energy Balance Transport Model . . . 3-17
Overview . . . 3-17
The Energy Balance Equations . . . 3-17
Density of States . . . 3-20
Energy Density Loss Rates . . . 3-20
Temperature Dependence of Relaxation Times . . . 3-21

Energy Dependent Mobilities 3-22
Boundary Physics 3-23
Overview 3-23
Ohmic Contacts 3-23
Schottky Contacts 3-24
Floating Contacts 3-26
Current Boundary Conditions 3-26
Insulating Contacts 3-27
Neumann Boundaries 3-27
Lumped R, L and C Elements 3-28
Distributed Contact Resistance 3-29
Energy Balance Boundary Conditions 3-30
Physical Models 3-31
Mobility Modeling 3-31
Overview 3-31
Low Field Mobility Models 3-31
Constant Mobility 3-32
Concentration Dependent Low Field Mobility Tables 3-32
The Analytic Low Field Mobility Model 3-34
The Arora Model for Low Field Mobility 3-35
The Carrier-Carrier Model Scattering for Low FIeld Mobility 3-36
Klaassen's Unified Low FIeld Mobility Models 3-38
Inversion Layer Moblity Models 3-43
Overview 3-43
The Lombardi CVT Model 3-44
The Yamaguchi Model. 3-46
The Tasch Model. 3-48
Perpendicular Electrid Field-Dependent Moblity Models 3-51
The Watt Model 3-51
Modifications to Watt Model 3-53
Shirahata's Mobility Model 3-54
Parallel Electric Field-Dependent Mobility 3-55
Carrier Temperature-Dependent Mobility 3-56
Mobility Model Summary 3-58
Carrier Generation-Recombination Models 3-60
Shockley-Read-Hall (SRH) Recombination 3-60
SRH Concentrarion Dependent Lifetiime Model 3-61
Klaassen's Concentration Dependent Lifetime Model. 3-61
Trap-Assisted Tunneling 3-62
Radiative (Direct) Recombination 3-64
Auger Recombination 3-64
Surface Recombination 3-65
Impact Ionization Models 3-66
Overview 3-66
Local Electric Field Models 3-67
Selberherr's Impact Ionization Model 3-67
Grant's Impact Ionization Model 3-69

Non-Local Energy Dependent Models 3-70
Crowell-Sze Impact Ionization Model 3-69
Toyabe Impact Ionization Model 3-71
Concannon's Impact Ionization Model 3-72
Band-to-Band Tunneling 3-74
Gate Current Models 3-75
Overview 3-75
Fowler-Nordheim Tunneling 3-75
Lucky Electron Hot Carrier Injection Model 3-76
Concannon's Injection Model 3-80
Device Level Reliability Modeling 3-83
Hansch MOS Reliability Model 3-83
ORCHID 3-84
The Ferroelectric Permittivity Model 3-84
Quantum Mechanical Models 3-85
Self Consistent Schrodinger Poisson Model 3-85
Quantum Moments Model 3-87
Quantum Correction Models 3-89
Hansch's Model 3-89
Van Dort's Model 3-89

Chapter 4:
S-PISCES 4-1

Introduction 4-1
Simulating Silicon Devices Using S-PISCES 4-1
Simulating MOS Technologies 4-1
Physical Models for MOSFETs 4-1
Meshing for MOS Devices 4-1
MOS Electrode Naming 4-3
Gate Workfunction 4-3
Interface Charge 4-4
Single Carrier Solutions 4-4
Energy Balance Solutions 4-4
ESD Simulation 4-4
Simulating Silicon Bipolar Devices 4-5
Physical Models for BJTs 4-5
Meshing Issues for BJTs 4-5
BJT Electrode Naming 4-5
Dual Base BJTs 4-5
Creating an Open Circuit Electrode 4-6
Solution Techniques for BJTs 4-6
Simulating Non-Volatile Memory Technologies (EEPROMs, FLASH Memories) 4-6
Defining Floating Gates 4-6
Gate Current Models 4-7
Gate Current Assignment (NEARFLG) 4-7
Simulating SOI Technologies 4-8
Meshing in SOI devices 4-8
Physical Models for SOI 4-8

Numerical Methods for SOI 4-9
SOI Physical Phenomena 4-9

Chapter 5:
BLAZE 5-1

Introduction 5-1
Basic Heterojunction Definitions 5-1
Alignment 5-2
The Affinity Rule 5-2
The ALIGN parameter on the MATERIAL statement 5-2
Manually Adjusting Material Affinity 5-3
EXAMPLE 1: 5-3
Using the ALIGN parameter on the MATERIAL statement 5-4
Manually Adjusting Material Affinity 5-4
EXAMPLE 2: 5-5
Manually Adjusting Material Affinity 5-6
EXAMPLE 3: 5-7
Manually Adjusting Material Affinity 5-9
EXAMPLE 4: 5-10
Using the affinity rule for the heterojunction 5-10
The Drift Diffusion Transport Model 5-12
Drift-Diffusion with Position Dependent Band Structure 5-12
The Thermionic Emission Transport Model 5-14
The Physical Models 5-14
Common Physical Models 5-15
Low Field Mobility Models 5-15
Parallel Electric Field-Dependent Mobility 5-16
Velocity Saturation with Energy Balance 5-17
Recombination and Generation Models 5-17
Material Dependent Physical Models 5-18
GaAs 5-18
Bandgap Narrowing 5-18
Low Field Mobilty 5-19
III-V and II-VI Materials 5-20
Al(x)Ga(1-x)As System 5-20
Bandgap 5-20
Electron Affinity 5-21
Density of States and Effective Mass 5-21
Dielectric Permittivity 5-22
Low Field Mobility 5-22
In(x)Ga(1-x)As(y)P(1-y) System 5-22
Bandgap 5-23
Electron Affinity 5-23
Density of States and Effective Mass 5-23
Dielectric Permittivity 5-24
Low Field Mobility 5-24
The Si(1-x)Ge(x) System 5-24
Bandgap 5-25

Electron Affinity 5-26
Density of States 5-26
Dielectric Function 5-26
Low Field Mobility 5-26
Velocity Saturation 5-26
Silicon Carbide (SiC) 5-27
Band Parameters for SiC 5-27
SiC Mobility Parameters 5-27
Isotropic Mobiliy 5-27
Anisotropic Mobility 5-28
Defining Anisotropic Mobility in ATLAS 5-28
Impact Ionization and Thermal Parameters 5-29
Simulating Heterojunction Devices with Blaze 5-29
Defining Material Regions with Positionaly Dependent Band Structure 5-29
Step Junctions 5-29
Graded Junctions 5-29
Defining Materials and Models 5-30
Materials 5-30
Models 5-30
Parser Functions 5-30

Chapter 6:
GIGA 6-1

Overview 6-1
Applications 6-1
Numerics 6-1
Physical Models. 6-2
The Lattice Heat Flow Equation 6-2
Specifying Heat Sink Layers For Thermal Solutions 6-2
Specifying Thermal Conductivity 6-2
Non-Isothermal Models 6-4
Effective Density Of States 6-4
Nonisothermal Current Densities 6-4
Heat Generation 6-5
Thermal Boundary Conditions 6-6
Temperature Dependent Material Parameters 6-7
C-interpreter Defined Peltier Coefficients 6-8
Applications of GIGA 6-8
Power Device Simulation Techniques 6-8
More Information 6-9

Chapter 7:
TFT 7-1

Polycrystalline and Amorphous Semiconductor Models 7-1
Introduction 7-1
Simulating TFT Devices 7-1
Defining The Materials 7-1
Defining The Defect States 7-1

Density of States Model. 7-2
Trapped Carrier Density 7-3
Steady-state Trap Recombination 7-4
Transient Traps 7-4
Trap-Assisted Tunneling 7-5
Continuous Defects 7-5
Discrete Defects 7-5
Plotting The Density Of States Versus Energy 7-6
Using the C-Interpreter to define DEFECTS 7-6
Setting Mobility and Other Models 7-7

Chapter 8:
Luminous 8-1

Introduction 8-1
Simulation Method 8-1
Ray Tracing 8-1
Defining the Incident Beam 8-1
Ray Splitting at Interfaces 8-2
Reflection and Transmission 8-3
Anti-Reflective Coatings 8-5
Light Absorption and Photogeneration 8-6
Photogeneration on a Non-uniform Mesh 8-7
Photogeneration at Contacts 8-7
User-Defined Arbitrary Photogeneration 8-7
Photocurrent and Quantum Efficiency 8-7
Simulating Photodetectors 8-9
Overview 8-9
Defining Optical Sources 8-9
Identifying an Optical Beam 8-9
Origin Plane of the Beam 8-9
Reflections 8-9
Monochromatic or Multispectral Sources 8-10
Defining Optical Properties of Materials 8-10
Setting Single Values For The Refractive Index 8-11
Setting A Wavelength Dependent Refractive Index 8-11
Extracting Dark Characteristics 8-12
Integrated Recombination 8-12
Extrapolation from High Temperatures 8-12
Numerical Solution Parameters 8-13
Extracting Detection Efficiency 8-13
Obtaining Quantum Efficiency versus Bias 8-14
Obtaining Transient Response to Optical Sources 8-14
Obtaining Frequency Response to Optical Sources 8-14
Obtaining Spatial Response 8-15
Obtaining Spectral Response 8-15
Simulating Solar Cells 8-16
Obtaining Open Circuit Voltage and Short Circuit Current 8-16
Simulating LEDs 8-16

Chapter 9: Laser 9-1

Introduction 9-1
Physical Models 9-1
Local Optical Gain 9-2
Stimulated Emission 9-4
Photon Rate Equations 9-4
Solution Techniques 9-6
Specifying Laser Simulation Problems 9-6
Numerical Parameters 9-7
Semiconductor Laser Simulation Techniques 9-8

Chapter 10: MIXEDMODE 10-1

Introduction 10-1
The MIXEDMODE Concept 10-1
Organization of this Chapter 10-1
Background 10-1
Advantages of MIXEDMODE Simulation 10-2
Using MIXEDMODE 10-2
Syntax Overview 10-2
General Syntax Rules 10-3
Circuit and Analysis Specification 10-3
Netlist Statements 10-3
Control Statements 10-5
Special statements 10-5
Device Simulation Syntax 10-6
Recommendations 10-6
Input Parsing 10-6
Scale and Suffixes 10-6
Numerics 10-7
Multi-Device Structure Representation 10-7
Extraction of Results 10-8
Using MIXEDMODE inside the VWF Automation Tools 10-9
Initial Settings 10-9
A Sample Command File 10-10
MIXEDMODE Syntax 10-12
A – ATLAS device to be simulated using device simulation 10-12
B – User-defined two-terminal element 10-13
C – Capacitor 10-13
D – Diode 10-13
E – Linear voltage controlled source 10-14
F– Linear current controlled current source 10-14
G – Linear voltage controlled current source 10-15
H – Linear current controlled voltage source 10-15
I – Independent current source 10-15
K – Coupling between two inductors 10-16
L – Inductor.Syntax 10-16

M – MOSFET 10-17
O – Optical source 10-17
Q – Bipolar junction transistor 10-18
R – Resistor 10-18
T – Lossless transmission line 10-19
V – Independent voltage source 10-19
.BEGIN 10-20
.END 10-20
.AC 10-20
.DC 10-20
.NET 10-21
.TRAN 10-23
.LOAD 10-23
.SAVE 10-24
.LOG 10-24
.IC 10-25
.NODESET 10-25
.NUMERIC 10-26
.OPTIONS 10-28
.MODEL 10-28
Transient Parameters 10-29
Overview 10-29
PULSE 10-29
EXP 10-30
GAUSS 10-30
SFFM 10-31
SIN 10-31
TABLE 10-32
AC Parameters 10-32
Compact Device Models 10-33
Diode Model 10-33
BJT Model 10-35
BJT Model Parameters 10-40
BJT Model Equations 10-48
DC Current Equations 10-49
Base Charge Equations 10-49
Variable Base Resistance Equations 10-50
Capacitance Equations 10-51
Temperature Effects Equations 10-54
Capacitance Temperature Equations 10-58
Parasitic Resistor Temperature Equations 10-60
MOSFET Model 10-60
DC Current Equations 10-64
User-Defined Two-Terminal Elements 10-68
Overview 10-68
User-Defined Model 10-68
Input Parameters 10-69

Output Parameters 10-69
Example 10-70

Bibliography

See Volume 2

Index

See Volume 2

This page intentionally left blank.

List of Figures

Figure No.	Caption Title	Page No.
2-1	ATLAS Inputs and Outputs	2-1
2-2	Examples Index in DeckBuild	2-4
2-3	ATLAS Command Groups with the Primary Statements in each Group	2-6
2-4	Non-uniform Mesh Creation using ATLAS Syntax	2-10
2-5	Analytical specification of a 2D Profile	2-12
2-6	Regrid on doping provides improved resolution of junction	2-14
2-7	Diagram showing syntax of Transient Voltage Ramp in ATLAS	2-35
3-1	Definition of the trap energy level for acceptor and donor traps, in reference to the conduction and valence band edges.	3-12
3-2	The lumped elements supported by ATLAS	3-28
4-1	Effect on MOS IV curve of progressive refinement of the vertical mesh spacing at the surface of the MOS channel	4-2
4-2	Effect of surface mesh spacing on simulated current for several MOS Mobility Models	4-3
5-1	Band Diagram of p-n heterojunction	5-1
5-2	Band diagram of heterojunction with band offset.	5-3
5-3	Band diagram of three material system (lowest Eg in center)	5-5
5-4	Band diagram of three material system (lowest Eg not in center)	5-7
5-5	Band diagram of heterojunction with Schottky contact	5-10
7-1	Syntax Guide to Define Two Tail States and Two Gaussian Distributions. NGA and NDG are the integrated values of the Gaussian distributions. Gaussians are entered on energies EGA and EGD respectively.	7-8
9-1	Optical Beam Geometry	9-2
9-2	Reflected and Transmitted Rays	9-2
9-3	Angles of incidence, reflection and transmission	9-3
9-4	Single Layer AR Coating Under Normal Incidence	9-6
10-1	Schematic of Primitive Example Circuit 10-5	
10-2	Display of a MIXEDMODE solution with two Numerical Devices	10-8
10-3	Diode Equivalent Circuit	10-34

Figure No.	Caption Title	Page No.
10-4	NPN BJT Current Convention (PNP are opposite)	10-38
10-5	BJT Equivalent Circuit for DC and Transient Analysis	10-39
10-6	A Lateral Transistor	10-48
10-7	A Vertical Transistor	10-49
10-8	Channel MOSFET Current Convention (p-channel currents are opposite).	10-62
10-9	MOSFET Equivalent Circuit for Transient Analysis	10-63

List of Tables

Figure No.	Table Title	Page No.
2-1	Carrier Statistics Models	2-23
2-2	Mobility Models	2-23
2-3	Recombination Models	2-24
2-4	Impact Ionization	2-25
2-5	Tunneling Models and Carrier Injection Models	2-25
2-6	Model Compatibility Chart	2-25
2-7	Parameter Syntax Replacements	2-30
3-1	User Definable Parameters for the Density of States	3-6
3-2	User Specifiable Parameters for Equation 3-36	3-7
3-3	User Definable Parameters of Slotbooms Bandgap Narrowing Model	3-8
3-4	User Specifiable Parameters for Equations 3-121 and 3-122	3-10
3-5	User Specifiable Parameters for Equations 3-51 to 3-57	3-13
3-6	User Specifiable Parameters for Equations 3-59- 3-61	3-14
3-7	User Specifiable Parameters for Trap-Assisted Tunneling Model	3-16
3-8	User Specifiable Parameters for Equations 3-95 and 3-96	3-21
3-9	User Specifiable Parameters for Variable Energy Relaxation Time.	3-21
3-10	User Specifiable Parameters for Equation 3-107	3-24
3-11	User Specifiable Parameters for Equations 3-108 to 3-109.	3-24
3-12	User Specifiable Parameters for Equations 3-110 to 3-111.	3-25
3-13	User Specifiable Parameters for Equation 3-114.	3-25
3-14	User Specifiable Parameters for Figure 3-1.	3-28
3-15	User Specifiable Parameters for Equation 3-117.	3-30
3-16	User Specifiable Parameters for the Constant Low Field Mobility Model	3-32
3-17	Mobility of Electrtons and Holes in Silicon at T=300K	3-32
3-18	User Specifiable Parameters for Equations 3-121 and 3-122	3-35
3-19	User Specifiable Parameters for Equations 3-123 and 3-124	3-36
3-20	User Specifiable Parameters for Equations 3-127 and 3-128..	3-37
3-21	User Specifiable Parameters for Equations 3-133 and 3-134	3-38
3-22	User Specifiable Parameters for Equation2 3-135 and 3-136	3-39
3-23	User Specifiable Parameters for Equations 3-137 and 3-138	3-40
3-24	User Specifiable Parameters for Equations 3-145 and 3-146	3-41
3-25	User Specifiable Parameters for Equations 3-147 and 3-148.	3-41
3-26	User Specifiable Parameters for Equations 3-149 and 3-150	3-42
3-27	User Specifiable Parameters for Equation 3-155 and 3-156	3-43
3-28	User Specifiable Parameters for Equations 3-158 to 3-163	3-45
3-29	User Specifiable Parameters for Equations 3-164 to 3-169.	3-47

Figure No.	Table Title	Page No.
3-30	Parameters for Equations 3-172 through 3-188.	3-50
3-31	User Specifiable Parameters for Equations 3-189 - 3-192	3-52
3-32	User Specifiable Parameters for Equations 3-195 and 3-196.	3-54
3-33	User Definable Parameters in the Field-Dependent Mobility Model	3-56
3-34	User Specifiable Parameters for Equations 3-197 - 3-207	3-58
3-35	User Specifiable Parameters for Equation 3-211.	3-60
3-36	User Specifiable Parameters for Equations 3-212 to 3-214.	3-61
3-37	User Specifiable Parameters for Equations 3-215 to 3-216.	3-62
3-38	User Specifiable Parameters for Equations 3-218 and 3-222.	3-63
3-39	User Specifiable Parameters for Equation 3-225.	3-64
3-40	User Specifiable Parameters for Equation 3-227 and 3-228.	3-65
3-41	User Specifiable Parameters for Equations 3-232 to 3-234.	3-66
3-42	User Definable Parameters in the Selberherr Impact Ionization Model.	3-67
3-43	Temperature Coefficient Parameters of the Selberherr Impact Ionization Model for Silicon in Equations 3-238 to 3-241.	3-68
3-44	Crowell-Sze Impact Ionization Model Parameters.	3-70
3-45	User Specifiable Parameters for Equations 3-258 - 3-261.	3-71
3-46	User Specifiable Parameters for Equations 3-262 and 3-263	3-73
3-47	User Definable Parameters for the Energy Distribution Functions	3-73
3-48	User Definable Parameters in the Band-toBand Tunneling Model	3-75
3-49	User Specifiable Parameters for Equation	3-76
3-50	User Definable Parameters in Concannon's Gate Current Model.	3-79
3-51	User Definable Parameters for Equations 3-305 and 3-306	3-83
3-52	User Specifiable Parameters for Equations 3-307 to 3-308.	3-85
3-53	Interpretations of `FIXED.FERMI` and `CALC.FERMI` Parameters during Post-Processed Schrodinger solution	3-87
5-1	User Specifiable Parameters for Equations 5-45 and 5-46	5-15
5-2	User Specifiable Parameters for Equation 5-47 and 5-48	5-16
5-3	User Specifiable Parameters for Equation 5-49 and 5-50	5-17
5-4	Default Bandgap Narrowing Values	5-18
5-5	Default Concentration Dependent Mobility for GaAs	5-19
5-6	6H-SiC Low Field Mobility	5-27
5-7	4H-SiC Low Field Mobility	5-27
6-1	User Specifiable Parameters for Equation 6-2	6-3
6-2	User Specifiable Parameters for Equation 6-3	6-3
6-3	User Specifiable Parameters for Equations 6-6 and 6-7	6-4
6-4	User Specifiable Parameters for Equations 6-14 and 6-15.	6-7
6-5	User Specifiable Parameters for Equations 6-16 and 6-17.	6-8

Figure No.	Table Title	Page No.

7-1 User Specifiable Parameters for Equations 7-2 to 7-5 7-2
7-2 User Specifiable Parameters for Equations 7-8 to 7-10 7-4
7-3 Additional Parameters for the DEFECTS Statement 7-6
8-1 User Specifiable Parameters for Equation 9-2 9-2
8-2 User Specifiable Parameters for Equation 9-3 9-3
8-3 User Specifiable Parameters for Equation 9-6 9-3
8-4 User Specifiable Parameters for Equation 9-7 9-4
8-5 User Specifiable Parameters for Equation 9-8 9-5
8-6 User Specifiable Parameters for LASER Loss Models 9-5
8-7 LASER Parameters 9-8
10-1 Diode DC Parameters 10-33
10-2 Diode Capacitance Parameters 10-34
10-3 A complete Gummel-Poon parameter list 10-39
10-4 Basic DC Model Parameters 10-40
10-5 Low Current Beta Degradation Parameters 10-40
10-6 Base Width Modulation Parameters 10-41
10-7 High Current Beta Degradation Effect Parameters 10-41
10-8 Parasitic Resistor Parameters 10-41
10-9 Junction Capacitor Parameters 10-42
10-10 Transit Time Parameters 10-42
10-11 Temperature Effect Parameters 10-43
10-12 MOSFET Model Parameters 10-62

[This page intentionally left blank]

Chapter 1: Introduction

Overview of ATLAS

ATLAS provides general capabilities for physically-based two, and three-dimensional simulation of semiconductor devices. ATLAS has a modular architecture that includes the following licensable tools and extensions:

- **ATLAS** Supplies general capabilities that are accessed by all the device simulation products.
- **S-PISCES** Simulates silicon devices.
- **BLAZE** Simulates devices fabricated using arbitrary semiconductors (including II-VI, III-V, and IV-IV materials), and heterojunction devices.
- **GIGA** Adds the ability to perform non-isothermal calculations that include the effects of lattice heating and heat sinks.
- **TFT** Supports the simulation of devices that are based on polycrystalline and amorphous materials.
- **LUMINOUS** Supplies capabilities required to simulate optoelectronic devices, including sophisticated ray-tracing.
- **LASER** Allows the simulation of heterostructure lasers by self-consistent solution of the Helmholtz equation for the optical field.
- **MIXEDMODE** Offers circuit simulation capabilities that employ numerical physically-based devices as well as compact analytical models.
- **QUANTUM** enables simulation of quantum effects through a quantum moments solver and schrodinger solver.
- **SiC** enables simualtion of anisotropic material and device models.
- **DEVICE3D** Provides capabilities for three-dimensional silicon device simulation.
- **BLAZE3D** Provides capabilities for three-dimensional compound semiconductor and heterojunction device simulation.
- **GIGA3D** Provides capabilities for non-isothermal three-dimensional device simulation.
- **MIXEDMODE3D** Provides capabilities for mixed circuit simulation and three-dimensional device simulation.
- **TFT3D** Provides capabilities for three-dimensional polycrystalline and amorphous semiconductor device simulation.
- **QUANTUM3D** Provides capabilities for three-dimensional quantum effects.
- **THERMAL3D** Provides capabilities for three-dimensional thermal analysis.
- **C-INTERPRETER** Allows inclusion of user-defined equations into device simulation calculations

ATLAS is designed to be used in conjunction with the VWF Interactive Tools. The VWF Interactive Tools, which include DECKBUILD, TONYPLOT, DEVEDIT, MASKVIEWS, and OPTIMIZER, are documented in the VWF INTERACTIVE TOOLS MANUAL.

The complete ATLAS documentation consists of the ATLAS User's Manual (this document) and the VWF INTERACTIVE TOOLS MANUAL.

How To Use This Manual

All users should read this Chapter and the Getting Started Chapter, in order to obtain an overview of ATLAS. Users of earlier versions of ATLAS will find it helpful to review the updated version history provided in Appendix D. The remaining chapters can then be referred to for a more detailed understanding of the capabilities of each ATLAS product.

If you are a new user, you should read the tutorial in the VWF INTERACTIVE TOOLS MANUAL to get a basic idea of the mechanics of using the user interface.

ATLAS is supplied with numerous examples that may be accessed conveniently through DeckBuild. The examples demonstrate most of the capabilities of ATLAS. The input files provided as part of these examples can provide an excellent starting point for developing your own input files.

Organization Of This Manual

The organization of this manual is as follows:

- **Chapter 1** provides a brief overview of ATLAS.
- **Chapter 2** helps all users to get an overivew of typical use.
- **Chapter 3** presents the base set of physical models available in ATLAS.
- **Chapter 4** documents the additional models and simulation capabilities of S-PISCES.
- **Chapter 5** documents the additional models and simulation capabilities of BLAZE.
- **Chapter 6** documents the models and simulation capabilities of GIGA.
- **Chapter 7** documents the models and simulation capabilities of TFT.
- **Chapter 8** documents the models and simulation capabilities of LUMINOUS.
- **Chapter 9** documents the models and simulation capabilities of LASER.
- **Chapter 10** documents the models and simulation capabilities of MIXEDMODE.
- **Chapter 11** documents the differences between the 3D capabilites in ATLAS as compared to the 2D products.
- **Chapter 12** documents the models and simulation capabilities of INTERCONNECT3D.
- **Chapter 13** documents the models and simulation capabilities of THERMAL3D.
- **Chapter 14** describes the numerical methods and options available.
- **Chapter 15** documents the input syntax .
- **Appendix A** explains how to use the C-interpreter to specify models.
- **Appendix B** summarizes the default numerical values of model coefficients used.
- **Appendix C** gives information on error messages.
- **Appendix D** provides a version history.

Technical Support

If you have difficulties or questions relating to the use of any Silvaco product, you can communicate with SILVACO support personnel by sending electronic mail to:

support@silvaco.com

When you send us an email message, please:

1. Explain the problem or question as fully as possible.
2. Include any input files that you have created and error messages that were generated
3. Provide us with the version number of ATLAS, the version numbers of the VWF INTERACTIVE TOOLS that you are using, and details of the hardware platform on which the problem was encountered.
4. Include your company affiliation, business telephone number and fax number.

You will be contacted promptly and your problem will be resolved as quickly as possible.

User feedback drives the further development of ATLAS. Please send your comments on the programs, suggestions for improvements, and additional feature requests to the electronic mail address given above.

Features And Capabilities of ATLAS

A Comprehensive Set of Models:

ATLAS provides a comprehensive set of physical models, including:

- DC, AC small-signal, and full time-dependency.
- Drift-diffusion transport models.
- Energy balance and Hydrodynamic transport models.
- Lattice heating and heatsinks.
- Graded and abrupt heterojunctions.
- Optoelectronic interactions with general ray tracing.
- Amorphous and polycrystalline materials.
- General circuit environments.
- Stimulated emission and radiation
- Fermi-Dirac and Boltzmann statistics.
- Advanced mobility models.
- Heavy doping effects.
- Full acceptor and donor trap dynamics
- Ohmic, Schottky, and insulating contacts.
- SRH, radiative, Auger, and surface recombination.
- Impact ionization (local and non-local).
- Floating gates.
- Band-to-band and Fowler-Nordheim tunneling.
- Hot carrier injection.
- Thermionic emission currents.

Fully Integrated Capabilities:

ATLAS works well with other software from SILVACO. For example, ATLAS

- Runs in the DECKBUILD interactive run-time environment.
- Is interfaced to TONYPLOT, the interactive graphics and analysis package.
- Accepts input from the ATHENA and SSUPREM3 process simulators.
- Is interfaced to UTMOST parameter extraction and device modeling software.
- can be used in experiments with the VWF AUTOMATION TOOLS.

Sophisticated Numerical Implementation

ATLAS uses powerful numerical techniques, including:

- Accurate and robust discretization techniques.
- Gummel, Newton, and block-Newton nonlinear iteration strategies.
- Efficient solvers, both direct and iterative, for linear subproblems.
- Powerful initial guess strategies.
- Small-signal calculation techniques that converge at all frequencies.
- Stable and accurate time integration.

Using ATLAS With Other Silvaco Software

ATLAS should only be used in conjunction with the VWF INTERACTIVE TOOLS. These include DECKBUILD, TONYPLOT, DEVEDIT, MASKVIEWS and OPTIMIZER. DECKBUILD provides an interactive run time environment. TONYPLOT supplies scientific visualization capabilities. DEVEDIT is an interactive tool for structure and mesh specification and refinement, and MASKVIEWS is an IC Layout Editor. The OPTIMIZER supports black box optimization across multiple simulators.

ATLAS is very often used in conjunction with the ATHENA process simulator. ATHENA predicts the physical structures that result from processing steps. The resulting physical structures are used as input by ATLAS, which then predicts the electrical characteristics associated with specified bias conditions. The combination of ATHENA and ATLAS makes it possible to determine the impact of process parameters on device characteristics.

The electrical characteristics predicted by ATLAS can be used as input by the UTMOST device characterization and SPICE modeling software. Compact models based on simulated device characteristics can then be supplied to circuit designers for preliminary circuit design. The use of ATHENA, ATLAS, UTMOST, and SMARTSPICE in combination makes it possible to predict the impact of process parameters on circuit characteristics.

ATLAS can be used as one of the simulators within the VWF AUTOMATION TOOLS. VWF makes it convenient to perform highly automated simulation-based experimentation. VWF is used in a way that closely mirrors experimental research and development procedures using split lots. It therefore links simulation very closely to technology development, resulting in significantly increased benefits from the use of simulation.

The Nature Of Physically-Based Simulation

ATLAS is a physically-based device simulator. Physically-based device simulation is not a familiar concept for all engineers. A brief overview is provided here to serve as a high-level orientation for new users.

Physically-based device simulators predict the electrical characteristics that are associated with specified physical structures and bias conditions. This is achieved by approximating the operation of a device onto a two or three dimensional grid, consisting of a number of grid points called nodes. By applying a set of differential equations, derived from Maxwells laws, onto this grid it is possible to simulate the transport of carriers through a structure. This means that the electrical performance of a device can now be modelled in DC, AC or transient modes of operation.

Physically-based simulation provides three major advantages: it is predictive, it provides insight, and it captures theoretical knowledge in a way that makes this knowledge available to non-experts.

Physically-based simulation is different from empirical modeling. The goal of empirical modeling is to obtain analytic formulae that approximate existing data with good accuracy and minimum complexity. Empirical models provide efficient approximation and interpolation. They do not provide insight, or predictive capabilities, or encapsulation of theoretical knowledge. Physically-based simulation is an alternative to experiments as a source of data.

Physically-based simulation has become very important for two reasons. First, it is almost always much quicker and cheaper than performing experiments. Second, it provides information that is difficult or impossible to measure. The drawbacks of simulation are that all the relevant physics must be incorporated into a simulator, and numerical procedures must be implemented to solve the associated equations. These tasks have been taken care of for users of ATLAS.

Users of physically-based device simulation tools must specify the problem to be simulated. Users of ATLAS specify device simulation problems by defining:

1. The physical structure to be simulated
2. The physical models to be used
3. The bias conditions for which electrical characteristics are to be simulated.

The subsequent chapters of this manual describe how to perform these steps.

Chapter 2:
Getting Started with ATLAS

Overview

ATLAS is a physically-based two and three dimensional device simulator. It predicts the electrical behavior of specified semiconductor structures, and provides insight into the internal physical mechanisms associated with device operation.

ATLAS can be used standalone or as a core tool in Silvaco's VIRTUAL WAFER FAB simulation environment. In the sequence of predicting the impact of process variables on circuit performance, device simulation fits between process simulation and SPICE model extraction.

If you are a new user, this chapter will help you to start using ATLAS effectively. The organization of topics parallels the steps that you go through to run the program. If you have used earlier versions of ATLAS you will find this chapter to be a useful overview of the new version, and a source of useful hints and advice.

Note: This chapter concentrates on the core functionality of ATLAS. If you are primarily interested in the specialized capabilities of particular ATLAS products you should read this chapter first, and then read the chapters that describe the ATLAS products you wish to use.

ATLAS Inputs and Outputs

Figure 2-1 shows the types of information that flow in and out of ATLAS. Most ATLAS simulations use two inputs: a text file that contains commands for ATLAS to execute, and a structure file that defines the structure that will be simulated.

ATLAS produces three types of output. The run-time output provides a guide to the progress of simulations running, and is where error messages and warning messages appear. Log files store all terminal voltages and currents from the device analysis, and solution files store two- and three-dimensional data relating to the values of solution variables within the device for a single bias point.

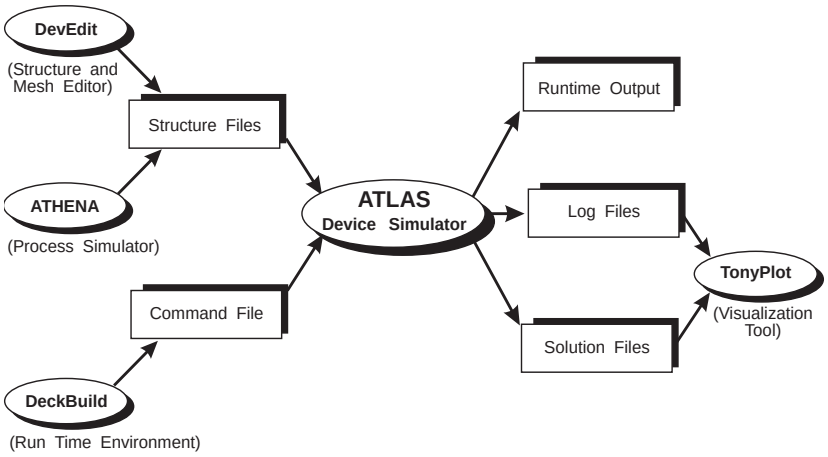

Figure 2-1: ATLAS Inputs and Outputs

Modes of Operation

ATLAS is normally used through the DECKBUILD run-time environment, which supports both interactive and batch mode operation. We strongly recommend that you always run ATLAS within DECKBUILD. In this section we present the basic information you need to run ATLAS in the DECKBUILD environment. The VWF INTERACTIVE TOOLS manual provides a more detailed description of the features and capabilities of DECKBUILD.

Interactive Mode With DeckBuild

To start ATLAS under DECKBUILD, type

```
deckbuild -as
```

at the UNIX system command prompt. The command line option -as instructs DECKBUILD to start ATLAS as the default simulator.

If you want to start from an existing input file you should start DECKBUILD by typing

```
deckbuild -as  <input filename>
```

The run-time output shows the execution of each ATLAS command and includes error messages, warnings, extracted parameters and other important output for evaluating each ATLAS run. When ATLAS is run interactively, run time output is sent to the output section of the DECKBUILD application window and can be saved as needed. You therefore do not need to save the run-time output explicitly. However, the following command line specifies the name of a file that will be used for storing the run-time output:

```
deckbuild -as  <input filename>  -outfile  <output filename>
```

In this case the run-time output is sent to the output file and to the output section of the DECKBUILD window.

Batch Mode With DeckBuild

To use DECKBUILD in a non-interactive or batch mode, you add the -run parameter to the command that invokes DECKBUILD. A pre-prepared command file is required for running in batch mode. It is advisable to save the run-time output to a file, since error messages in the run-time output would otherwise be lost when the batch job completes.

```
deckbuild -run -as  <input filename>  -outfile  <output filename>
```

Using this command requires a local X-Windows system to be running. The job runs inside a DECKBUILD icon on the terminal, and quits automatically when the ATLAS simulation is complete.

You can also run DECKBUILD using a remote display:

```
deckbuild -run -as <input file> -outfile <output file> -display<hostname>:0.0
```

No Windows Batch Mode With DeckBuild

For completely non-X Windows operation of DECKBUILD, the -ascii parameter is required:

```
deckbuild -run -ascii -as <input filename> -outfile <output filename>
```

This command directs DECKBUILD to run the ATLAS simulation without any display of the DECKBUILD window or icon.This is useful for remote execution without an X windows emulator or for replacing Unix-based ATLAS runs within framework programs.

When using batch mode use the UNIX command suffix '&' to detach the job from the current command shell. To run a remote ATLAS simulation under DECKBUILD without display, and then logout from the system, use the UNIX nohup command before the DeckBuild command line:

```
nohup deckbuild -run -ascii -as <input filename> -outfile <output filename> &
```

Running ATLAS inside Deckbuild

Each ATLAS run inside DECKBUILD should start with the line:

```
go atlas
```

A single input file may contain several ATLAS runs each separated with a `go atlas` line. Input files within DECKBUILD can also contain runs from other programs such as ATHENA or DEVEDIT along with the ATLAS runs.

Running a given version number of ATLAS

The go statement can be modified to provide parameters for the ATLAS run. To run version 4.3.0.R the syntax is:

```
go atlas simflags="-V 4.3.0.R"
```

Starting Parallel ATLAS

The -P option is used to set the number of processors to use in a parallel **ATLAS** run. If the number set by -P is greater than the number of processors available or than the number of parallel thread licenses, the number is automatically reduced to this cap number. To run on 4 processors:

```
go atlas simflags="-V 4.3.2.C -P 4"
```

Batch Mode Without DeckBuild

It is possible to run ATLAS outside the DECKBUILD environment. However this is not recommended by Silvaco. Users who do not want the overhead of the DECKBUILD window can use the "**No Windows Mode**" described above. Many important features such as variable substitution, automatic interfacing to process simulation, and parameter extraction, are not available outside the DECKBUILD environment. To run ATLAS directly under UNIX use the command:

```
atlas <input filename>
```

To save the run-time output to a file, do not use the UNIX redirect command (>), simply specify the name of the output file:

```
atlas <input filename> -logfile  <output filename>
```

Note: The standard examples supplied with ATLAS will not run correctly outside of DECKBUILD.

Accessing The Examples

ATLAS is supplied with more than 300 standard examples that demonstrate the way that the program is used to simulate many different technologies. The examples are instructional and it is strongly recommended that new users use these examples as a starting point for creating their own simulations. One of the first things you should learn is how to access, load and run these examples.

The examples are accessed from the menu system in DECKBUILD. To select and load an example:

1. Start DECKBUILD with ATLAS as the simulator as described in the previous section.
2. Pull down the **MainControl** menu using the right hand mouse button. There are options on this menu for **MainControl, Optimizer, Examples, Help** etc.
3. Select **Examples**. An index will appear in a DECKBUILD**: Examples** window (see below)

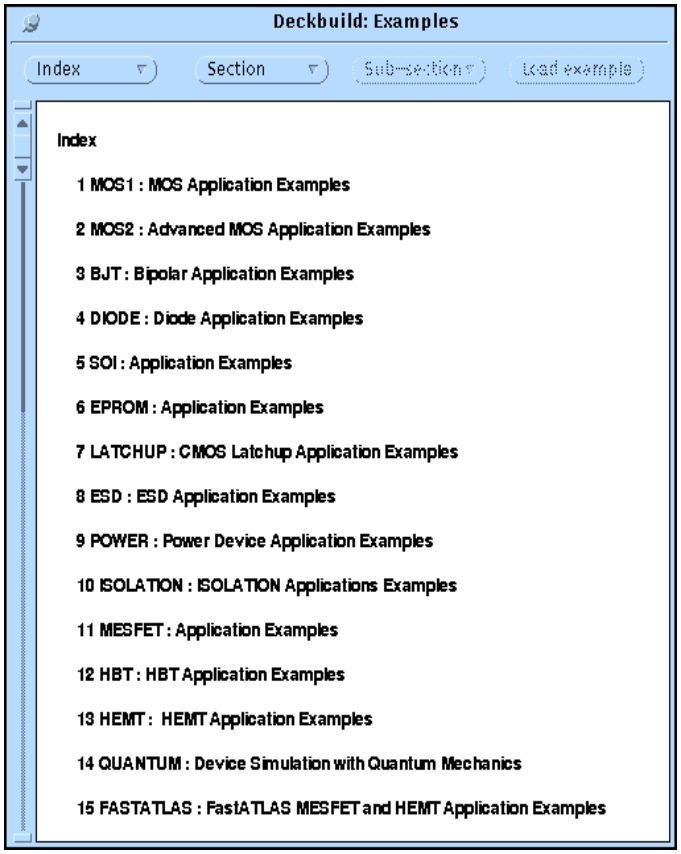
Deckbuild: Examples
Index
Section
Sub-section
Load example
Index
1 MOS1 : MOS Application Examples
2 MOS2 : Advanced MOS Application Examples
3 BJT : Bipolar Application Examples
4 DIODE : Diode Application Examples
5 SOI : Application Examples
6 EPROM : Application Examples
7 LATCHUP : CMOS Latchup Application Examples
8 ESD : ESD Application Examples
9 POWER : Power Device Application Examples
10 ISOLATION : ISOLATION Applications Examples
11 MESFET : Application Examples
12 HBT : HBT Application Examples
13 HEMT : HEMT Application Examples
14 QUANTUM : Device Simulation with Quantum Mechanics
15 FASTATLAS : FastATLAS MESFET and HEMT Application Examples

Figure 2-2: Examples Index in DeckBuild

The examples are divided by technology or technology group. The most common technologies are clear (e.g., MOS, BJT) while others are grouped with similar devices (e.g., IGBT and LDMOS are under POWER, and solar cell and photodiode are under OPTOELECTRONICS).

4. Choose the technology you are interested in by double clicking the left mouse button over that item in the examples index.
5. A list of examples for that technology will appear. These examples typically illustrate different devices, applications, or types of simulation.

   An examples search feature also exists by pressing the right hand mouse button over the button **Index**. Wildcards can be used in the search.
6. Choose a particular example by double clicking the left mouse button over that item in the list.
7. A text description of the example will appear in the window. This text describes the important physical mechanisms in the simulation, as well as details of the ATLAS syntax used. You should read this information before proceeding.
8. Press the button **Load Example.** The input command file for the example will be copied into your current working directory together with any associated files. A copy of the command file will be loaded into DECKBUILD**.** (Note that the **Load Example** button remains faded out until step 6 is performed correctly).
9. To run the example, press the **Run** button in the middle frame of the DECKBUILD application window.
10. Alternatively, most examples are supplied with results that are copied into the current working directory along with the input file. To view the results, select (highlight) the name of the results file and select the DECKBUILD menu option **Tools-Plot**. Details on the use of TONYPLOT can be found in the VWF INTERACTIVE TOOLS manual.

The ATLAS Syntax

An ATLAS command file is a list of commands for ATLAS to execute. This list is stored as an ASCII text file that can be prepared in DECKBUILD or using any text editor. Preparation of the input file in DECKBUILD is preferred, and can be made easier by appropriate use of the DECKBUILD Commands menu.

Statements and Parameters

The input file contains a sequence of statements. Each statement consists of a keyword that identifies the statement and a set of parameters. The general format is:

```
<STATEMENT>    <PARAMETER>=<VALUE>
```

With very few exceptions the input syntax is not case sensitive. One important exception is that commands described in this manual as being executed by DECKBUILD rather than ATLAS are case sensitive. This includes `EXTRACT`, `SET`, `GO`, and `SYSTEM`. In addition filenames for input and output under UNIX are case sensitive.

For any `<STATEMENT>` ATLAS may have four different types for the `<VALUE>` `parameter`: real, integer, character and logical.

An example of a statement line is:

```
DOPING UNIFORM N.TYPE CONCENTRATION=1.0e16 REGION=1 OUTFILE=my.dop
```

The statement is `DOPING`. All other items are parameters of the `DOPING` statement. `UNIFORM` and N.TYPE are logical parameters. Their presence on the line sets their values to true, otherwise they take their default values (usually false). `CONCENTRATION` is a real parameter and takes floating point numbers as input values, `REGION` is an integer parameter taking only integer numbers as input and `OUTFILE` is a character parameter type taking strings as input.

1. The statement keyword must come first, but after this the order of parameters within a statement is not important.
2. It is only necessary to use enough letters of any parameter to distinguish it from any other parameter on the same statement. Thus CONCENTRATION can be shortened to `CONC`. However `REGION` cannot be shortened to `R` since there is also a parameter RATIO associated with the DOPING statement.
3. Logicals can be explicitly set to false by preceding them with the ^ symbol.
4. Any line beginning with a # is ignored. These lines are used as comments.
5. ATLAS can read up to 256 characters on one line. However, it is best to spread long input statements over several lines to make the input file more readable. The character \ at the end of a line indicates continuation.

The Order of ATLAS Commands

The order in which statements occur in an ATLAS input file is important. There are five groups of statements, and these must occur in the correct order. These groups are indicated in Figure 2-3.

Each input file must contain these five groups in order. Failure to do this will usually cause an error message and termination of the program, but it could lead to incorrect operation of the program. For example, material parameters or models set in the wrong order may not be used in the calculations. The order of statements within the mesh definition, structural definition, and solution groups is also important.

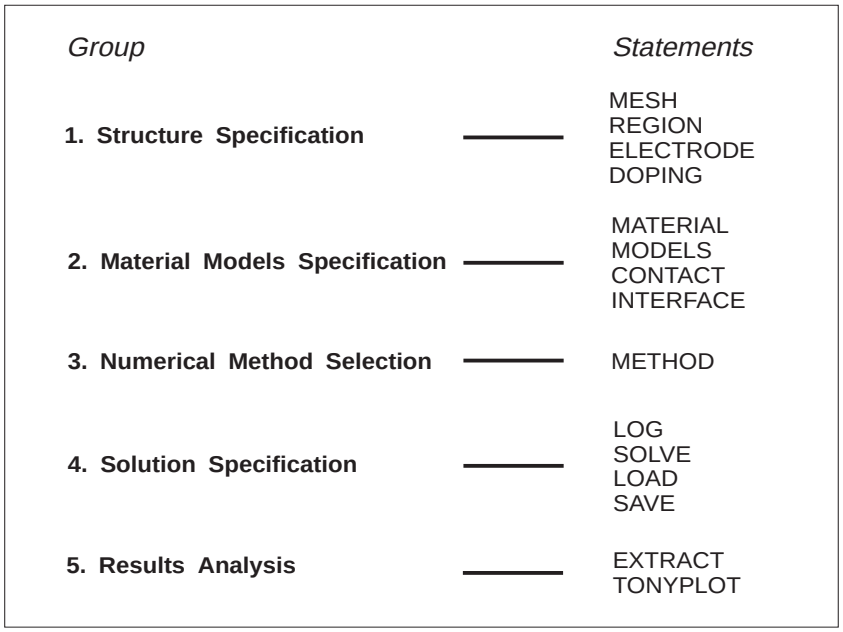

Figure 2-3: ATLAS Command Groups with the Primary Statements in each Group

The DeckBuild Command Menu

The DECKBUILD **Command Menu** can help you to create input files. This menu is found under the **Commands** button on DECKBUILD's main screen. The **Commands** menu is configured for ATLAS whenever ATLAS is the currently active simulator in DECKBUILD. (When ATLAS is active, this is indicated in the lower bar of the DECKBUILD application window, and an ATLAS command prompt appears in the DECKBUILD output section).

The DECKBUILD **Command Menu** gives you access to pop-up windows in which you type information. When you select the "Write" button, syntactically correct statements are written to the DECKBUILD text edit region. The DECKBUILD **Command Menu** does not support all possible ATLAS syntax, but aims to cover the most commonly used commands.

Quick Start for PISCES-II Users

This section is to provide quickstart instructions for users who may be familiar with the syntax and use of the Stanford University PISCES-II program or other device simulators derived from this program. The major differences between ATLAS and PISCES-II are:

- all graphics are handled by a separate interactive graphics program, TONYPLOT. The PISCES-II graphics commands `PLOT.1D`, `PLOT.2D`, `CONTOUR`, `VECTOR` etc. are not required.Using TONYPLOT it is no longer necessary to run the device simulator simply to plot or alter graphics.
- there is no need to separate individual ATLAS simulations into separate input files. Multiple runs of ATLAS are possible in the same input file separated by the line `go atlas`. Similarly there is no need to separate process and device simulation runs of Silvaco products into separate input files. A single file containing ATHENA and ATLAS syntax is permitted in DECKBUILD.
- the interface from process to device simulation is handled though a single file format compatible with other programs. The file read by ATLAS is the default output file format of ATHENA. No special file format for the interface is required.
- when defining a grid structure within ATLAS the `NODE` and `LOCATION` syntax to define exact grid line numbers in X and Y is not recommended. A more reliable and easier to use syntax using `LOCATION` and `SPACING` is available.
- use of the `REGRID` command is not recommended due to the creation of obtuse triangles. A standalone program DEVEDIT can be used as a grid pre-processor for ATLAS.
- all numerical method selection commands and parameters are on the `METHOD` statement. The `SYMBOLIC` statement is not used. Typically `SYMBOLIC` and `METHOD` were used as a coupled pair of statements and it is more convienient to use a single statement. Most of the old parameters of the `SYMBOLIC` statement have the same meaning and names despite this move to a single statement. One notable change in ATLAS is that numerical methods can be combined together. There is a separate section later in this chapter concerning translation of PISCES-II numerics statements.
- various general purpose commands are actually part of the DECKBUILD User Environment. These include `SET`, `EXTRACT`, `GO`, `SYSTEM`, and `SOURCE`. These commands can be interspersed inside ATLAS syntax.
- Variable substitution is supported for both numerical and string variables using the `SET` statement and $ symbol. To avoid confusion the # symbol is prefered to the $ symbol for comment statements.

In addition to these changes the physical models are in general different in ATLAS. Most of the original PISCES-II models have been preserved but often are not the default or the recommended models to use. Consult the on-line examples for technology specific information about models.

Defining A Structure

A device structure can be defined in three different ways in ATLAS.

1. An existing structure can be read in from a file. The structure can have been created by an earlier ATLAS run or by another program such as ATHENA or DEVEDIT. A single statement loads in the mesh, geometry, electrode positions and DOPING of the structure. This statement is:

   ```
   MESH INFILE=<filename>
   ```

2. The input structure can be transferred from ATHENA or DEVEDIT through the automatic interface feature of DECKBUILD.
3. A structure can be constructed using the ATLAS command language.

The first and second methods are more convenient than the third and are to be preferred whenever possible.

Interface From ATHENA

When ATHENA and ATLAS are run under DECKBUILD users can take advantage of an automatic interface between the two programs. Use the following steps to load the complete mesh, geometry and DOPING from ATHENA to ATLAS:

1. Deposit and pattern electrode material in ATHENA
2. Use the ELECTRODE statement in ATHENA to define contact positions. Specify the x and y coordinates as cross-hairs to pin-point a region. The whole region is then turned into electrode. In many cases only the x coordinate is needed.

   ```
   ELECTRODE NAME=gate X=1.3 [Y=-0.1])
   ```

3. There is a special case to specify a contact on the bottom of the structure.

   ```
   ELECTRODE NAME=substrate BACKSIDE
   ```

4. Save a structure file while ATHENA is still the active simulator.

   ```
   STRUCTURE OUTF=nmos.str
   ```

5. Start ATLAS with the command '*go atlas*'. This will automatically load the most recent structure from ATHENA into ATLAS.

Note: Do not specify a `MESH` command in ATLAS.

If you subsequently need to load the structure saved in step 4 into ATLAS without using the auto-interface capability, use the MESH command. (eg. MESH INF=nmos.str).

ATLAS inherits the grid used most recently by ATHENA. With a careful choice of initial mesh or by using the grid manipulation techniques in ATHENA it is possible to produce a final mesh from ATHENA that will give good results in ATLAS. However, a grid that is appropriate for process simulation is not always appropriate for device simulation. If the final ATHENA mesh is not appropriate for ATLAS, DEVEDIT may be used to re-mesh the structure, or the `REGRID` command may be used.

Interface From DevEdit

A 2D or 3D structure created by DEVEDIT can be read into ATLAS using the command:

```
MESH INF=<structure filename>
```

This single statement loads in the mesh, geometry, electrode positions and doping of the structure. ATLAS will automatically determine whether the mesh is 2D for S-PISCES or BLAZE or 3D for DEVICE3D or BLAZE3D.

If the structure coming from DEVEDIT was originally created by ATHENA, the electrodes should be defined in ATHENA as described in the previous section. If the structure is created in DEVEDIT the electrode regions should be defined on the **Region/Add** menu in DEVEDIT.

Using The Command Language To Define A Structure

To define a device through the ATLAS command language you must first define a mesh. This mesh or grid covers the physical simulation domain. The mesh is defined by a series of horizontal and vertical lines and the spacing between them. Next, regions within this mesh are allocated to different materials as required to construct the device. For example, the specification of a MOS device requires the specification of silicon and silicon dioxide regions. After the regions are defined, the location of electrodes is specified. The final step is to specify the doping in each region.

When using the command language to define a structure, the information described in the following four sub-sections must be specified in the order listed.

Specifying The Initial Mesh

The first statement must be

```
MESH SPACE.MULT=<VALUE>
```

This is followed by a series of X.MESH and Y.MESH statements

```
X.MESH LOCATION=<VALUE> SPACING=<VALUE>
.
Y.MESH LOCATION=<VALUE> SPACING=<VALUE>
.
```

The SPACE.MULT parameter value is used as a scaling factor for the mesh created by the X.MESH and Y.MESH statements. The default value is 1. Values greater than 1 will create a globally coarser mesh for fast simulation. Values less than 1 will create a globally finer mesh for increased accuracy. The X.MESH and Y.MESH statements are used to specify the locations in microns of vertical and horizontal lines, respectively, together with the vertical or horizontal spacing associated with that line. At least two mesh lines must be specified for each direction. ATLAS automatically inserts any new lines required to allow for gradual transitions in the spacing values between any adjacent lines. The X.MESH and Y.MESH statements must be listed in the order of increasing x and y. Both negative and positive values of x and y are allowed. Figure 2-4 illustrates how these statements work. On the left hand plot, note how the spacing of the vertical lines varies from 1 mm at x=0 and x=10 mm to 0.5 mm micron at x=5 mm. On the right hand plot, note how specifying the parameter SPACE.MULT to have a value of 0.5 has doubled the density of the mesh in both the x- and y-directions.

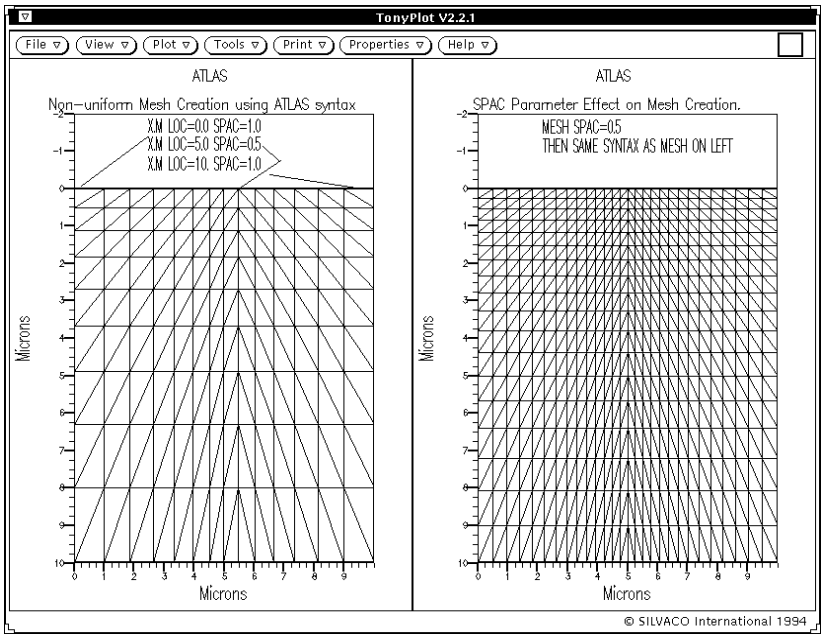

Figure 2-4: Non-uniform Mesh Creation using ATLAS Syntax

After an initial mesh has been defined, you can remove grid lines in specified regions. This is typically done in regions of the device for which a coarse grid is expected to be sufficient such as the substrate. The removal of grid lines is accomplished using the ELIMINATE statement. The ELIMINATE statement removes every second mesh line in the specified direction from within a specified rectangle. The statement

```
ELIMINATE COLUMNS X.MIN=0 X.MAX=4 Y.MIN=0.0 Y.MAX=3
```

removes every second vertical grid line within the rectangle bounded by x=0, x=4, y=0 and y=3 microns.

Specifying Regions And Materials

After the mesh is specified, every part of the mesh must be assigned to be a particular material type. This is done with REGION statements:

```
REGION number=<integer> <material_type> <position parameters>
```

Region numbers must start at 1 and are increased for each subsequent region statement. Up to 55 different regions are allowed in ATLAS. A large number of materials is available. If a composition-dependent material type is defined, the x- and y-composition fractions can also be specified on the REGION statement.

The position parameters are specified in microns using the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters. If the position parameters of a new statement overlap those of a previous region statement, the overlapped area is assigned as the material type of the new region. Care must be taken to ensure that materials are assigned to all mesh points in the structure. If this is not done, error messages will appear and ATLAS will fail to run.

You can use the MATERIAL statement to specify the material properties of the defined regions. However the complete mesh and doping definition must be completed before any MATERIAL statements can be used. The specification of material properties is described in the later section "Specifying Material Properties".

Cylindrical Coordinates

Cylindrical coordinates are often used when simulating discrete power devices. In this mode, ATLAS operates with x=0 as the axis of symmetry around which the cylindrical geometry is placed. Many of the default units change when cylindrical coordinates are used. The calculated current is in Amps rather than the usual Amps per micron. External elements are specified in absolute units (e.g., Farads, not Farads/micron for capacitors).

The MESH statement must be used to specify cylindrical symmetry. The following statement creates a mesh which contains cylindrical symmetry. There are 20 mesh nodes along the x-axis and 20 mesh nodes along the y-axis.

```
MESH NX=20 NY=20 CYLINDRICAL
```

The following statement imports a mesh which contains cylindrical symmetry.

```
MESH INF=mesh0.str CYLINDRICAL
```

Note: The CYLINDRICAL parameter setting is not stored in mesh files. Therefore the parameter must be specified each time a mesh file which contains cylindrical symmetry is loaded.

Specifying Electrodes

After you have specified the regions and materials you must define at least one electrode that contacts a semiconductor material. This is done with the ELECTRODE statement:

```
ELECTRODE NAME=<electrode name> <position_parameters>
```

Up to 50 electrodes may be specified. The position parameters are specified in microns using the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters. Multiple electrode statements may have the same electrode name. Nodes that are associated with the same electrode name are treated as being electrically connected.

Some shortcuts may be used when defining the location of an electrode. If no y-coordinate parameters are specified, the electrode is assumed to be located on the top of the structure. In addition, you may use the parameters RIGHT, LEFT, TOP, and BOTTOM. For example:

```
ELECTRODE NAME=SOURCE LEFT LENGTH=0.5
```

specifies the source electrode starts at the top left corner of the structure and extends to the right for the distance LENGTH.

Specifying Doping

You can specify analytical doping distributions, or have ATLAS read in profiles that come from either process simulation or experiment. You specify the doping using the DOPING statement:

```
DOPING <distribution_type> <dopant_type> <position_parameters>
```

Analytical Doping Profiles

Analytical doping profiles can have uniform or Gaussian forms. The parameters defining the analytical distribution are specified on the DOPING statement. Two examples are shown below, and their combined effect is shown in Figure 2-5. The first DOPING statement specifies a uniform n-type doping density of 10^{16} cm^{-3} in the region that was previously labelled as region #1.

```
DOPING UNIFORM CONCENTRATION=1E16 N.TYPE REGION=1
```

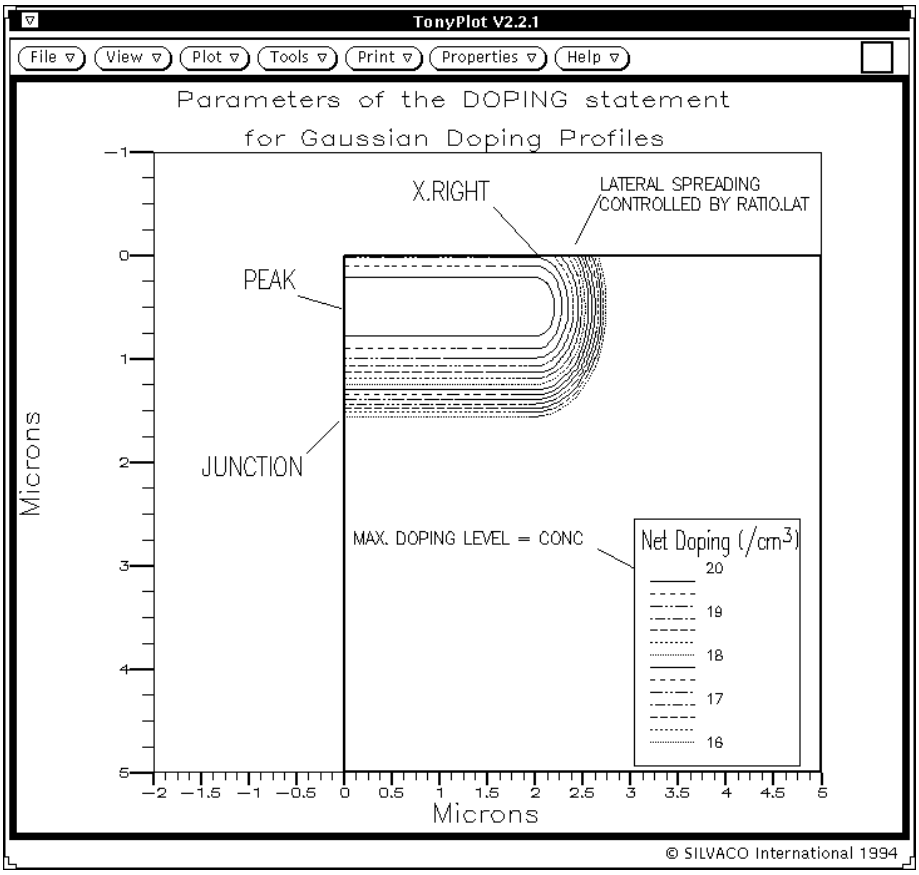

Figure 2-5: Analytical specification of a 2D Profile

Position parameters X.MIN, X.MAX, Y.MIN, and Y.MAX may be used instead of a region number. The second statement specifies a p-type Gaussian profile with a peak concentration of 10^{18} cm^{-3}.

```
DOPING GAUSSIAN CONCENTRATION=1E18 CHARACTERISTIC=0.05 P.TYPE \
       X.LEFT=0.0 X.RIGHT=1.0 PEAK=0.1
```

This statement specifies that the peak doping is located along a line from x = 0 to x = 1 microns. Perpendicular to the peak line the doping drops off according to a Gaussian distribution with a standard deviation of 0.05 mm. At x < 0 or x > 1 the doping drops off laterally with a default standard deviation that is 70% of CHARACTERISTIC.. This lateral roll-off can be altered with the RATIO.LATERAL parameter. If a Gaussian profile is being added to an area that was already defined with the opposite dopant type then the JUNCTION parameter may be used to specify the position of the junction depth instead of specifying the standard deviation using the CHARACTERISTIC parameter .

Importing 1D SSUPREM3 Doping Profiles

One-dimensional doping profiles can be read into ATLAS from a SSUPREM3 output file. The doping data must have been saved from SSUPREM3 using the statement

```
STRUCTURE OUTFILE=<output filename>
```

at the end of the SSUPREM3 run.

In ATLAS, the MASTER parameter of the doping statement specifies that a SSUPREM3 file will be read by ATLAS. Since this file will usually contain all the dopants from the SSUPREM3 simulation, the desired dopant type must also be specified. The statement

```
DOPING MASTER INFILE=mydata.dat BORON REGION=1
```

specifies that the boron profile from the file mydata.dat should be imported and used in region #1. SSUPREM3 profiles are imported into ATLAS one at a time,i.e., one doping statement is used for each profile or dopant. The statements

```
DOPING MASTER INFILE=mydata.dat BORON OUTFILE=doping.dat
DOPING MASTER INFILE=mydata.dat ARSENIC X.RIGHT=0.8 RATIO=0.75
DOPING MASTER INFILE=mydata.dat ARSENIC X.LEFT=2.2 RATIO=0.75
```

offset the arsenic doping from boron to create a 2D doping profile from a single SSUPREM3 result. It is advisable to include the OUTFILE parameter on the first doping statement to create a 2D doping file. This file will be used in the next section to interpolate doping on a refined mesh after a REGRID. It cannot be plotted in TONYPLOT. The position parameters and RATIO.LATERAL are used in the same manner as for analytical doping profiles to set the extent of the 1-D profile.

Remeshing Using The Command Language

When specifying a structure using the command language, it can be difficult to define a suitable grid. The main problem is that the meshes required to resolve 2-D doping profiles and curved junctions are quite complicated and simple rectangular meshes require an excessive number of nodes to resolve such profiles. If a device structure only includes regions of uniform doping, there is usually no need to regrid. However, when realistic 2-D doping profiles are present, a regrid may be necessary

---

Note: The recommended solution for defining complex mesh structures for ATLAS is to use the standalone program DEVEDIT.

Regrid On Doping

ATLAS includes a regridding capability that generates a fine mesh only in a localized region. You specify a quantity on which the regrid is to be performed. the mesh is then refined in regions where the specified quantity varies rapidly. Whenever a specified quantity (usually doping) changes quickly, the regridding will automatically grade the mesh accordingly. A regrid on doping can be obtained before any solutions are obtained. This can be accomplished with the statement.

```
REGRID LOGARITHM DOPING RATIO=2 SMOOTH.KEY=4 DOPFILE=<filename1> \
       OUTFILE=<filename2>
```

This statement must be used after the MESH, REGION, MATERIAL, ELECTRODE, and DOPING statements described previously. The effects of this REGRID statement on a simple Diode structure are shown in Figure 2-6. In this statement, regridding will be done in such a way that the new mesh will resolve doping profiles to two orders of magnitude in change.

The doping file '*filename1*' must have been specified on the first DOPING statement with the OUTFILE parameter. The results of the regrid are saved in the file '*filename2*'. The SMOOTH.KEY parameter value selects a smoothing algorithm. A value of 4 is typically best as this algorithm tends to produce the fewest obtuse triangles. For a complete description of the various smoothing algorithms, refer to Chapter 14 which describes numerical techniques.

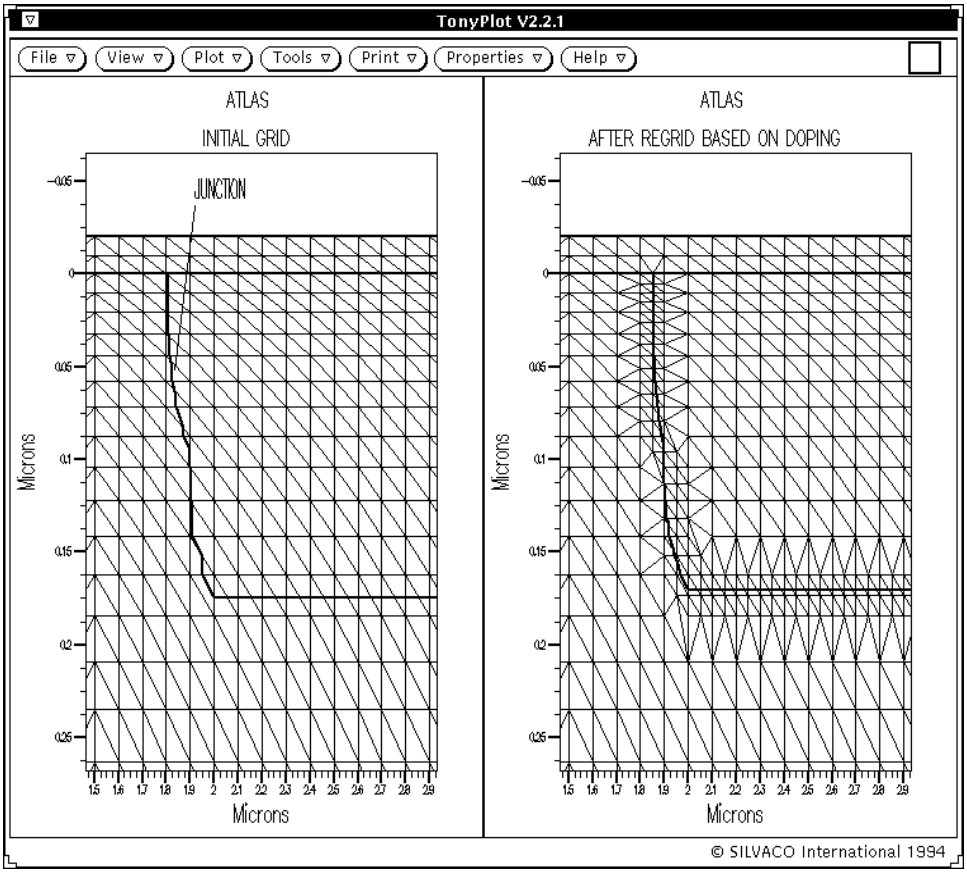

Figure 2-6: Regrid on doping provides improved resolution of junction

Regrid Using Solution Variables

The REGRID statement can use a wide range of solution variables as the basis for mesh refinement. Regrid on potential is often used for high voltage power devices. Note that regrid on solution variables can only be used after a solution has already been obtained. After a regrid on a solution variable, the solution must be re-solved at the same bias in ATLAS.

For example, to regrid on potential and re-solve at the original bias, the following statements are used:

```
REGRID POTENTIAL RATIO=0.2 MAX.LEVEL=1 SMOOTH.K=4 DOPFILE=<filename1>
SOLVE PREV
```

Note: The REGRID statement may be used any number of times on a structure. However it often advisable to quit and restart ATLAS between regrids on electrical quantities. The `go atlas` statement can be used to do this. This should be followed by a MESH statement loading the output file of the REGRID command and a re-setting of all material and model parameters.

Specifying 3D Structures

The syntax for forming 3D device structures is an extension of the 2D syntax described in the previous section. The MESH statement should appear as:

```
MESH THREE.D
```

The parameter THREE.D tells ATLAS that a three dimensional grid will be specified. The other statements used to specify 3-D structures and grids are the same as for 2-D with the addition of z-direction specifications. The statements

```
MESH THREE.D X.MESH LOCATION=0 SPACING=0.15
X.MESH LOCATION=3 SPACING=0.15
Y.MESH LOCATION=0 SPACING=0.01
Y.MESH LOCATION=0.4 SPACING=0.01
Y.MESH LOCATION=3 SPACING=0.5
Z.MESH LOCATION=0 SPACING=0.1
Z.MESH LOCATION=3 SPACING=0.1
```

define a 3D mesh that is uniform in the x and z directions and varies in the y direction.

Position parameters for the z-direction (Z.MIN and Z.MAX) are also used on REGION, ELECTRODE, or DOPING statements.

General Comments Regarding Grids

Specifying a good grid is a crucial issue in device simulation. There is a trade-off between the requirements of accuracy and numerical efficiency. Accuracy requires a fine grid that resolves the structure in solutions. Numerical efficiency is greater when fewer grid points are used. The critical areas to resolve are difficult to generalize since they depend on the technology and the transport phenomena. The only generalization possible is that most critical areas tend to coincide with reverse-biased metallurgical junctions. Typical critical areas are:

- High electric fields at the drain/channel junction in MOSFETs
- The transverse electric field beneath the MOSFET gate
- Recombination effects around the emitter/base junction in BJTs
- Areas of high impact ionization
- Around heterojunctions in HBT's, HEMTs.

The cpu time required to obtain a solution is typically proportional to N^{α} where N is the number of nodes and α varies from 2 to 3 depending on the complexity of the problem. Thus it is most efficient to allocate a fine grid only in critical areas, and a coarser grid elsewhere.

The three most important factors to look for in any grid are:

- Ensure adequate mesh density in high field areas
- Avoid obtuse triangles in the current path or high field areas
- Avoid abrupt discontinuities in mesh density

More information concerning grids is provided in the Numerical Techniques chapter.

Maximum Number Of Nodes

ATLAS sets some limits on the maximum number of grid nodes that may be used.However this should not be viewed as users as a bottleneck to achieving simulation results. In the default version:

- 2-D ATLAS simulations have a maximum node limit of 9,600.
- 3-D ATLAS simulations have an upper limit of 200,000 nodes with no more than 20,000 in any one plane.

This limit is high enough that for almost all simulations of conventional devices, running out of nodes is never an issue. For most 2-D simulations, accurate results can be obtained with somewhere between two thousand and four thousand node points properly located in the structure.

If the node limits are exceeded, error messages will appear and ATLAS will not run successfully. Decreasing the mesh density is the first option since simulations with the maximum nodes will take an extremely long time to complete. However, if it is deemed absolutely necessary to include more than the maximum number of nodes, please contact your local Silvaco office to describe your needs so that Silvaco may be better able to meet them. A version of each ATLAS release with 20,000 nodes is routinely available. Versions of 2-D ATLAS up to 100,000 nodes have been distributed.

A node point limitation below these values might be seen due to virtual memory constraints on your hardware. For each simulation, ATLAS dynamically allocates the virtual memory. See the Silvaco Installation Guide for information about virtual memory requirements. The virtual memory used by the program depends not only on the number of nodes but also on such items as the models used and the number of equations that are solved.

Defining Material Parameters And Models

After the mesh, geometry, and doping profiles are defined, you can modify the characteristics of electrodes, change the default material parameters, and choose which physical models ATLAS will use during the device simulation. These actions are accomplished using the CONTACT, MATERIAL, and MODELS statements respectively. Impact ionization models can be enabled using the IMPACT statement, and interface properties are set by the INTERFACE statement. Many parameters are accessible through the Silvaco C-Interpreter, which is described in more detail in Appendix A. This allows you to define customized equations for some models.

Specifying Contact Characteristics

Workfunction for Gates or Schottky Contacts

An electrode in contact with semiconductor material is assumed by default to be ohmic. If a work function is defined, the electrode is treated as a Schottky contact. The CONTACT statement is used to specify the metal workfunction of one or more electrodes. The NAME parameter is used to identify which electrode will have its properties modified.

The WORKFUNCTION parameter sets the workfunction of the electrode. The statement

```
CONTACT NAME=gate WORKFUNCTION=4.8
```

sets the workfunction of the electrode named gate to 4.8eV. The workfunctions of several commonly used contact materials may be specified using the name of the material. Workfunctions for ALUMINUM, N.POLYSILICON, P.POLYSILICON, TUNGSTEN, and TU.DISILICIDE may be specified in this way. The following statementsets the workfunction for a n-type polysilicon gate contact.

```
CONTACT NAME=gate N.POLYSILICON
```

Aluminum contacts on heavily doped silicon is usually ohmic, and you should not specify a workfunction for this situation. For example, for MOS devices you should not specify

```
CONTACT NAME=drain ALUMINUM    /* wrong */
```

The CONTACT statement may also be used to specify barrier and dipole lowering of the Schottky barrier height. Barrier lowering is enabled by specifying the BARRIER parameter while dipole lowering is specified using the ALPHA parameter. The statement

```
CONTACT NAME=anode WORKFUNCTION=4.9 BARRIER ALPHA=1.0e-7‘
```

sets the work function of the Schottky contact named anode to 4.9eV, enables barrier lowering, and sets the dipole lowering coefficient to 1 nm.

Note: When a Schottky barrier is defined at a contact it is recommended that a fine y-mesh is present just beneath the contact, inside the semiconductor. This allows the Schottky depletion region to be accurately simulated.

Setting Current Boundary Conditions

The CONTACT statement is also used to change an electrode from voltage control to current control. Current controlled electrodes are useful when simulating devices where the current is highly sensitive to voltage or is a multivalued function of voltage (e.g. post-breakdown and when there is snap-back).

The statement

```
CONTACT NAME=drain CURRENT
```

changes the drain electrode to current control. The BLOCK or NEWTON solution methods arequired for all simulations using a current boundary condition.

Defining External Resistors, Capacitors or Inductors

Lumped resistance, capacitance, and inductance connected to an electrode can be specified using the RESISTANCE, CAPACITANCE, and INDUCTANCE parameters on the CONTACT statement. The statement

```
CONTACT NAME=drain RESISTANCE=50.0 CAPACITANCE=20e-12 INDUCTANCE=1e-6
```

specifies a parallel resistor and capacitor of 50 ohms and 20 pF respectively in series with a 1 μH inductor. Note that in 2D simulations these passive element values are scaled by the width in the third dimension. Since in 2D ATLAS assumes a 1μm width, the resistance becomes 50 Ω-μm.

Distributed contact resistance for an electrode can be specified using the CON.RESIST parameter. The statement

```
CONTACT NAME=source CON.RESISTANCE=0.01
```

specifies that the source contact has a distributed resistance of 0.01 Ωcm^2.

Note: Simulations with external resistors, capacitors or inductors must be solved using the BLOCK or NEWTON solution method

---

Floating Contacts

The CONTACT statement is also used to define a floating electrode. There are two distinctly different situations for which floating electrodes are important. The first is **floating gate** electrodes used in EEPROM and other programmable devices. The second is for **contacts directly onto semiconductor materials** such as floating field plates in power devices.

Floating gates are enabled by specifying the parameter FLOATING on the CONTACT statement. The statement

```
CONTACT NAME=fgate FLOATING
```

specifies that the electrode named fgate will be floating and that charge boundary conditions will apply.

For contacts directly onto semiconductor the FLOATING parameter cannot be used. This type of floating electrode is best simulated by specifying current boundary conditions on the CONTACT statement. The statement

```
CONTACT NAME=drain CURRENT
```

specifies current boundary conditions for drain electrode. On subsequent SOLVE statements the drain current boundary condition will default to zero current hence floating the contact.

It is also possible make a floating contact to a semiconductor using a very large resistor attached to the contact instead

```
CONTACT NAME=drain RESIST=1e20
```

Note that extremely large resistance values must be used to keep the current through the contact insignificant bearing in mind the tolerance on potential will allow the contact voltage to move slightly above zero. For example if the tolerance is 10^{-5}V and the defined resistance was only 10MΩμm then a current of 10^{-12} A/μm may flow through the contact which is probably significant in breakdown simulations.

Shorting two contacts together

It is possible in ATLAS to tie two or more contact together so that voltages on both contacts are equal. This is useful for many technologies for example dual base bipolar transistors. There are several methods for achieving this depending on how the structure was initial defined.

If the structure is defined using ATLAS syntax it is possible to have multiple ELECTRODE statements with the same NAME parameter defining separate locations within the device structure. In this case the areas defined to be electrodes will be considered as having the same applied voltage. A single current will appear combining the current through both ELECTRODE areas.

Similarly if two separate metal regions in ATHENA are defined using the ATHENA ELECTRODE statement to have the same name, then in ATLAS these two electrodes will be considered as shorted together.

If the electrodes are defined with different names the following syntax canbe used to link the voltages applied to the two electrodes.

```
CONTACT NAME=base1 COMMON=base
.
SOLVE VBASE=0.1
```

Here the electrode *base1* will be linked to the electrode *base*. Later the applied 0.1V on *base* will also appear on *base1*. However ATLAS will calculate and store **separate** currents for both *base* and *base1*. This can be a useful feature. However in some cases such as where functions of the currents are required in EXTRACT or TONYPLOT it is undesirable. The parameter SHORT may be added to the CONTACT statement above to specify that only a single *base* current will appear combining the currents from *base* and *base1*.

When loading a structure from ATHENA or DEVEDIT where two defined electrode regions are touching, ATLAS will automatically short these and use the electrode name that was defined first.

Making an open circuit contact

It is often required to perform a simulation with an open circuit on one of the defined electrodes. From a device simulation viewpoint there are three different methods that will accomplish this. These are;

- Entirely deleting an electrode from the structure file.
- Adding an extremely large lumped resistance, for example $10^{20}\Omega$, onto the contact to be made open circuited.
- Switching the boundary conditions on the contact to be made open circuited from voltage controlled to current controlled and then specifying a very small current through that electrode.

Each of these methods are feasible but if in doing so a floating region is created within the structure then numerical convergence may be affected. As a result it is normally recommended that the second method, that of a lumped resistance, be used as it ensures better convergence.

Specifying Material Properties

Semiconductor, Insulator, or Conductor

All materials are split into three classes: semiconductors, insulators and conductors. Each class requires a different set of parameters to be specified. For semiconductors these properties include electron affinity, band gap, density of states and saturation velocities. There are default parameters for material properties used in device simulation for many materials. Appendix B lists default material parameters and includes a discussion on the differences between specifying parameters for semiconductors, insulators and conductors.

Setting Parameters

The MATERIAL statement allows you to specify your own values for these basic parameters. Your values can apply to a specified material or a specified region. The statement

```
MATERIAL MATERIAL=Silicon EG300=1.12 MUN=1100
```

sets the band gap and low field electron mobility in all silicon regions in the device. If the material properties are defined by region, the region is specified using the REGION or NAME parameter on the MATERIAL statement. For example, the statement

```
MATERIAL REGION=2 TAUN0=2e-7 TAUP0=1e-5
```

sets the electron and hole Shockley-Read-Hall recombination lifetimes for region number two. If the name "base" has been defined using the NAME parameter in the REGION statement, then the statement

```
MATERIAL NAME=base NC300=3e19
```

sets the conduction band density of states at 300 K for the region named base.

The description of the MATERIAL statement in the Statements Chapter provides a complete list of all the material parameters that are available.

Heterojunction Materials

The material properties of heterojunctions can also be modified with the MATERIAL statement. In addition to the regular material parameters, compositionally dependent material parameters can be defined. These include compositionally dependent band parameters, dielectric constants, saturation velocities and so on.

For heterojunction material systems, the bandgap difference between the materials is divided between conduction and valence bands. The ALIGN parameter specifies the fraction of this difference that is applied to the conduction band edge. This determines the electron and hole barrier height and overrides any electron affinity specification. The statement

```
MATERIAL MATERIAL=InGaAs ALIGN=0.36
MATERIAL MATERIAL=InP    ALIGN=0.36
```

specifies that 36% of the band gap difference between InGaAs and InP goes to the conduction band and 64% goes to the valence band. If for example, the band gap difference (Δ Eg) for this material system is 0.6 eV, then the conduction band barrier height for this example is 0.216 eV and the valence band barrier height it 0.384 eV.

For heterojunction devices, the transport models may be different for each material. These models and their coefficients may be specified for each material using the MODEL statement. Refer to the section "Specifying Physical Models" for a description of this option.

Specifying Interface Properties

The INTERFACE statement is used to define the interface charge density and surface recombination velocity at interfaces between semiconductors and insulators . For example, the statement

```
INTERFACE QF=3e10
```

specifies that all interfaces between semiconductors and insulators have a fixed charge of $3.10^{10}cm^{-2}$. In many cases, the interface of interest is restricted to a specific region. This can be accomplished with the X.MIN, X.MAX, Y.MIN, and Y.MAX parameters on the INTERFACE statement. These parameters define a rectangle within which the interface properties apply. The statement

```
INTERFACE QF=3e10 X.MIN=1.0 X.MAX=2 Y.MIN=0.0 Y.MAX=0.5
```

restricts the interface charge to the semiconductor-insulator boundary within the specified rectangle. In addition to fixed charge, surface recombination velocity and thermionic emission are enabled and defined with the INTERFACE statement. A detailed description of the INTERFACE statement is given in the Statements Chapter.

Specifying Physical Models

Physical models are specified using the MODELS and IMPACT statements. Parameters for these models appear on many statements including MODELS, IMPACT, MOBILITY and MATERIAL. The physical models can be grouped into five classes: mobility, recombination, carrier statistics, impact ionization, and tunneling. The Physics chapter contains details of each model. Tables 2.1 through 2.5 give summary descriptions and recommendations on the use of each model. Table 2.6 is a guide for compatibility between models.

All models with the exception of impact ionization are specified on the *MODELS* statement. Impact ionization is specified on the IMPACT statement. The statement

```
MODELS CONMOB FLDMOB SRH FERMIDIRAC
IMPACT SELB
```

specifies that the standard concentration dependent mobility, parallel field mobility, Schockley-Read-Hall recombination with fixed carrier lifetimes, Fermi Dirac statistics and Selberherr impact ionization models should be used.

ATLAS also provides an easy method for selecting the correct models for various technologies. The MOS, BIP, PROGRAM, and ERASE parameters for the MODELS statement configure a basic set of mobility, recombination, carrier statistics, and tunneling models. The parameters MOS and BIP enable the models for MOSFET and bipolar devices while PROGRAM and ERASE enable the models for programming and erasing programmable devices. For example, the statement

```
MODELS MOS PRINT
```

enables the CVT, SRH, and FERMIDIRAC models while the statement

```
MODELS BIPOLAR PRINT
```

enables the CONMOB, FLDMOB, CONSRH, AUGER and BGN.

Note: The PRINT parameter lists to the run time output the models and parameters which will be used during the simulation. This allows the user to verify models and material parameters. It is highly recommend to include the PRINT parameter on the MODEL statement.

Physical models can be enabled on a material by material basis. This is useful for heterojunction device simulation and other simulations where multiple semiconductor regions are defined and may have different characteristics. For example, the statements

```
MODEL MATERIAL=GaAs FLDMOB EVSATMOD=1 ECRITN=6.0e3 CONMOB
MODEL MATERIAL=InGaAs SRH FLDMOB EVSATMOD=1 \
ECRITN=3.0e3
```

change both the mobility models, and critical electric field used in each material. For devices based on advanced materials, these model parameters should be investigated carefully.

Energy Balance Models

The conventional drift-diffusion model of charge transport neglects "non-local" effects such as velocity overshoot and reduced energy dependent impact ionization. ATLAS can model these effects through the use of an energy balance model, which uses a higher order approximation of the Boltzmann transport equation. In this formalism, transport parameters such as mobility and impact ionization are functions of the local carrier temperature rather than the local electric field.

To enable the energy balance transport model, the HCTE, HCTE.EL, and HCTE.HO parameters on the MODELS statement are used. These parameters enable the energy transport model for both carriers, electrons only , or holes only respectively. The statement

```
MODELS MOS HCTE
```

enables the energy balance transport model for both electrons and holes in addition to the default MOSFET models.

Summary Of Physical Models

Table 2-1. Carrier Statistics Models		
Model	**Syntax**	**Notes**
Boltzmann	BOLTZMANN	Default model
Fermi-Dirac	FERMI	Reduced carrier concentrations in heavily doped regions (statistical approach)
Incomplete Ionization	INCOMPLETE	Accounts for dopant freeze-out. Typically used at low temperatures.
Silicon Ionization Model	IONIZ	Accounts for full ionization for heavily doped Si. Use with INCOMPLETE.
Bandgap Narrowing	BGN	Inportant in heavily doped regions. Critical for bipolar gain. Use Klaassen Model.

Table 2-2. Mobility Models		
Model	**Syntax**	**Notes**
Concentration Dependent	CONMOB	Lookup table valid at 300K for Si and GaAs only. Uses simple power law temperature dependence.
Concentration and Temperature Dependent	ANALYTIC	Caughey-Thomas formula. Tuned for 77-450K.
Arora’s Model	ARORA	Alternative to ANALYTIC for Si
Carrier-Carrier Scattering	CCSMOB	Dorkel-Leturq Model. Includes n, N and T dependence.. Important when carrier concentration is high (e.g., forward bias power devices).
Parallel Electric Field Dependence	FLDMOB	Si and GaAs models. Required to model any type of velocity satuation effect.
Tasch Model	TASCH	Includes transverse field dependence. Only for planar devices. Needs very fine grid.
Watt Model	WATT	Transverse field model applied to surface nodes only.

Table 2-2. Mobility Models		
Model	**Syntax**	**Notes**
Klaassen Model	KLA	Includes N, T and n dependence. Applies separate mobility to majority and minority carriers. Recommended for bipolar devices
Shirahata Model	SHI	Includes N, $E_{\perp}$. An alternative surface mobility model that can be combined with KLA.
Modified Watt	MOD.WATT	Extension of WATT model to non-surface nodes. Applies constant $E_{\perp}$ effects. Best model for planar MOS devices
Lombardi (CVT) Model	CVT	Complete model including N, T, E// and $E_{\perp}$ effects. Good for non-planar devices.
Yamaguchi Model	YAMAGUCHI	Includes N, E// and $E_{\perp}$ effects. Only for 300K.

Table 2-3. Recombination Models		
Model	**Syntax**	**Notes**
Shockley-Read-Hall	SRH	Uses fixed minority carrier life-times. Should be used in most simu-lations.
Concentration Dependent	CONSRH	Uses concentration dependent life-times. Recommended for Si.
Auger	AUGER	Direct transition of three carriers. Important at high current densities.
Optical	OPTR	Band-band recombination. For direct materials only.
Surface	S.N S.P	Recombination at semiconductor to insulator interfaces. Set on the INTERFACE statement.

Table 2-4. Impact Ionization		
Model	**Syntax**	**Notes**
Silberrherr's Model	IMPACT SELB	Recommended for most cases. Includes temperture dependent parameters.
Grant's Model	IMPACT	Similiar to Selberrherr's model but with different coefficients.
Crowell-Sze	IMPACT CROWELL	Uses dependence on carrier scattering length.
Toyabe Model		Non-local model used with Energy Balance. Any IMPACT syntax is accepted.
Concannon	N.CONCAN P.CONCAN	Non-local model developed in Flash EEPROM technologies.

Table 2-5. TUnneling Models and Carrier Injection Models		
Model	**Syntax**	**Notes**
Fowler-Nordheim (electrons)	FNORD	Self-consistent calculation of tunneling through insulators. Used in EEPROMs.
Fowler-Nordheim (holes)	FNHOLES	As FNORD for holes. Not usually required.
Band-to-Band (standard)	BBT.STD	For direct transitions. Required with very high fields.
Klaassen Band-to-Band	BBT.KL	Includes direct and indirect transitions.
Hot Electron Injection	HEI	Models energetic carriers tunneling throug insulators. Used for gate current and Flash EEPROM programming.
Hot Hole Injection	HHI	As HEI for holes. For ULSI PMOS devices.
Concannon Gate Current Model	N.CONCAN P.CONCAN	Non-local gate model consistent with Concannon substrate current model.

Table 2-6. Model Compatibility Chart

	CONMOB	BLDMOB	TFLDMB2	YAMAGUCHI	CVT	ARORA	ANALYTIC	CCSMOB	SURFACE	LATTICE H	E.BALANCE
CONMOB [CM]	—	OK	OK	YA	CV	AR	AN	CC	OK	OK	OK

Table 2-6. Model Compatibility Chart (Continued)

	CONMOB	BLDMOB	TFLDMB2	YAMAGUCHI	CVT	ARORA	ANALYTIC	CCSMOB	SURFACE	LATTICE H	E.BALANCE
FLDMOB [FM]	OK	—	TF[1]	YA	CV	OK	OK	OK	OK	OK	OK
TFLDMB2 [TF]	OK	TF[1]	—	YA	CV	OK	OK	TF	TF	OK	OK
YAMAGUCHI [YA]	YA	YA	YA	—	CV	YA	YA	YA	YA	NO	NO
CVT [CV]	CV	CV	CV	CV	—	CV	CV	CV	CV	OK	OK
ARORA [AR]	AR	OK	OK	YA	CV	—	AR	CC	OK	OK	OK
ANALYTIC [AN]	AN	OK	OK	YA	CV		—	CC	OK	OK	OK
CCSMOB [CC]	CC	OK	TF	YA	CV	CC	CC	—	OK	OK	OK
SURFMOB [SF]	OK	OK	TF	YA	CV	OK	OK	OK	—	OK	OK
LATTICE H [LH]	OK	OK	OK	NO	OK	OK	OK	OK	OK	—	OK
E.BALANCE [EB]	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	[2]

Key To Table Entries

MODEL ABBREVIATION = The model that supercedes when a combination is specified. In some cases, but not all, a warning message i s issued when a model is ignored.

OK = This combination is allowed

NO = This combination is not allowed

NOTES

1 Uses internal model similar to FLDMOB

2 When models including a parallel electric field dependence are used with energy balance the electric field term is replaced by a function of carrier temperature.

Using The C-Interpreter To Specify Models

One of the ATLAS products is a C language interpreter that allows you to specify many of the models used by ATLAS. To use these functions, the user must implement the model in C as equations in a special file called an ATLAS lib file. The default ATLAS template file can be accessed by typing

```
atlas -T <filename>
```

at the UNIX command prompt. This creates a default template file with the name specified by the <*filename*> parameter. A listing of the default C-Interpreter functions can be found in the description section for the various ATLAS statements such as MATERIALS and MOBILITY. To use the interpreter functions, the corresponding parameters must be given on the statements with the name of the C language file containing the model given as the parameter value. For example, the statement

```
MATERIAL NAME=Silicon F.MUNSAT=munsat.lib
```

specifies that the file *munsat.lib* contains the C-Interpreter function for the specification of the parallel field dependent electron mobility model.

Choosing Numerical Methods

Numerical Solution Techniques

Several different numerical methods can be used for calculating the solutions to semiconductor device problems. Different solution methods are optimum in different situations and some guidelines will be given here. Full details of the numerical solution techniques can be found in the Chapter on Numerical Techniques.Numerical methods are given on the METHOD statements of the input file.

Different combinations of models will require ATLAS to solve up to six equations. For each of the model types there are basically three types of solution techniques: (a) de-coupled (GUMMEL), (b) fully coupled (NEWTON) and (c) BLOCK. In simple terms, the de-coupled technique like the Gummel method will solve for each unknown in turn keeping the other variables constant, repeating the process until a stable solution is achieved. Fully coupled techniques such as the Newton method solve the total system of unknowns together. The combined or block methods will solve some equations fully coupled, while others are de-coupled.

In general, the Gummel method is useful where the system of equations is weakly coupled, but has only linear convergence. The Newton method is useful when the system of equations is strongly coupled and has quadratic convergence. The Newton method may however spend extra time solving for quantities which are essentially constant or weakly coupled. Newton also requires a more accurate initial guess to the problem to obtain convergence. Thus, a block method can provide for faster simulations times in these cases over Newton. Gummel can often provide better initial guesses to problems. It can be useful to start a solution with a few Gummel iterations to generate a better guess and then switch to Newton to complete the solution. Specification of the solution method is carried out as follows:

```
METHOD GUMMEL BLOCK NEWTON
```

The exact meaning of the statement depends upon the particular models it is applied to. This will be discussed in the following sections.

Basic Drift Diffusion Calculations

The isothermal drift diffusion model requires the solution of three equations for potential, electron concentration and hole concentration. Specifying GUMMEL or NEWTON alone will produce simple Gummel or Newton solutions as detailed above. For almost all cases the Newton method is preferred and it is the default.

Specifying:

```
METHOD GUMMEL NEWTON
```

will cause the solver to start with Gummel iterations and then switch to Newton, if convergence is not achieved. This is a very robust, although more time consuming way of obtaining solutions for any device. However this method is highly recommended for all simulations with floating regions such as SOI transistors. A floating region is defined as an area of doping which is separated from all electrodes by a pn junction.

BLOCK is equivalent to NEWTON for all isothermal drift-diffusion simulations.

Drift Diffusion Calculations with Lattice Heating

When the lattice heating model is added to drift diffusion an extra equation is added. The BLOCK algorithm solves the three drift diffusion equations as a Newton solution and follows this with a decoupled solution of the heat flow equation. The NEWTON algorithm solves all four equations in a coupled manner. NEWTON is preferred once the temperature is high, however BLOCK is quicker for low temperature gradients. Typically the combination used is:

```
METHOD BLOCK NEWTON
```

Energy Balance Calculations

The energy balance model requires the solution of up to 5 coupled equations. GUMMEL and NEWTON have the same meanings as with the drift diffusion model (i.e. GUMMEL specifies a decoupled solution and NEWTON specifies a fully coupled solution).

However, BLOCK performs a coupled solution of potential, carrier continuity equations followed by a coupled solution of carrier energy balance, and carrier continuity equations.

It is possible to switch from BLOCK to NEWTON by specifying multiple solution methods on the same line. For example:

```
METHOD BLOCK NEWTON
```

will begin with BLOCK iterations then switch to NEWTON if convergence is still not achieved. This is the most robust approach for many energy balance applications.

The points at which the algorithms switch is predetermined, but can also be changed on the METHOD statement. The default values set by Silvaco work well for most circumstances.

Energy Balance Calculations with Lattice Heating

When non-isothermal solutions are performed in conjunction with energy balance models, a system of up to six equations must be solved. GUMMEL or NEWTON solve the equations iteratively or fully coupled respectively. BLOCK initially performs the same function as with energy balance calculations, then solves the lattice heating equation in a de-coupled manner.

Setting The Number Of Carriers

ATLAS can solve both electron and hole continuity equations, or only for one or none. This choice can be made using the parameter CARRIERS. For example,

```
METHOD CARRIERS=2
```

specifies a solution for both carriers is required. This is the default. With one carrier the parameter ELEC or HOLE is needed. For example, for hole solutions only:

```
METHOD CARRIERS=1 HOLE
```

To select a solution for potential only specify:

```
METHOD CARRIERS=0
```

Note: Setting the number of carriers uing the syntax MODEL NUMCARR=<n> is obsolete and should not be used

---

Important Parameters Of the METHOD Statement

It is possible to alter all of the parameters relevant to the numerical solution process. This is not recommended unless you have expert knowledge of the numerical algorithms. All of these parameters have been assigned optimal values for most solution conditions. It is beyond the scope of this chapter to give more details. Further information can be found in the Numerical Methods Chapter.

Two parameters, however, are worth noting at this stage:

1. CLIMIT or CLIM.DD specify minimal values of concentrations to be resolved by the solver. Sometimes it can be necessary to reduce this value to aid solutions of breakdown characteristics. A value of CLIMIT=1e-4 is recommended for all simulations of breakdown where the pre-breakdown current is small. CLIM.DD is equivalent to CLIMIT but uses the more convienient units of cm^{-3} for the critical concentration.
2. DVMAX controls the maximum update of potential per iteration of Newton's method. The default corresponds to 1V. For power devices requiring large voltages an increased value of DVMAX might be needed. DVMAX=1e8 can improve the speed of high voltage bias ramps.
3. CLIM.EB controls the cut-off carrier concentration below which the program will not consider the error in the carrier temperature. This is applied in energy balance simulations to avoid excessive calculations of the carrier temperature at locations in the structure where the carrier concentration is low. However if this parameter is set to too high so that the carrier temperature errors for significant carrier concentrations are being ignored, unpredictable and mostly incorrect results will be seen.

Restrictions on the Choice of METHOD

The following cases all require METHOD NEWTON CARRIERS=2 to be set for isothermal drift-diffusion simulations. Both BLOCK and/or NEWTON are permitted for lattice heat and energy balance:

- current boundary conditions
- distributed or lumped external elements
- AC analysis
- impact ionization

Note: Simulations using the GUMMEL method in these cases may lead to non-covergence or incorrect results

---

Pisces-II Compatibility

Previous releases of ATLAS (2.0.0.R) and other PISCES-II based programs, use the SYMBOLIC command to define the solution method and the number of carriers to be included in the solution. In this version of ATLAS, the solution method is specified completely on the METHOD statement.

The COMB parameter which was available in earlier ATLAS versions is no longer required, as it is replaced by either the BLOCK method or the combination of GUMMEL and NEWTON parameters. The following table identifies direct translations of old syntax to new.

Note: These are direct translations and not necessarily the best choices of numerical methods.

Table 2-7. Parameter Syntax Replacements

Old Syntax (V2.0.0.R)	New Syntax
`symbolic newton carriers=2`	`method newton`
`symbolic newton carriers=1 elec`	`method newton carriers=1 electron`
`symbolic gummel carriers=0`	`method gummel carriers=0`
`symbolic newton carriers=2` `method comb`	`method gummel newton`

`models lat.temp`
`symbolic newton carriers=2`
`method comb`
	`models lat.temp`
	`method block`

`models hcte`
`symbolic gummel carriers=2`
`method comb`
	`models hcte`
	`method block`

Obtaining Solutions

ATLAS can calculate DC, AC small signal, and transient solutions. Obtaining solutions is rather analogous to setting up parametric test equipment for device tests. You usually define the voltages on each of the electrodes in the device. ATLAS then calculates the current through each electrode. ATLAS also calculates internal quantities, such as carrier concentrations and electric fields throughout the device. This is information that is difficult or impossible to measure.

In all simulations the device starts with zero bias on all electrodes. Solutions are obtained by stepping the biases on electrodes from this initial equilibrium condition. As will be discussed, due to the initial guess strategy, voltage step sizes are limited. This section concentrates on defining solution procedures. Saving results using the LOG or SAVE statements, and analysing and displaying these results, is in the subsequent section.

DC Solutions

In DC solutions, the voltage on each electrode is specified using the SOLVE statement. For example, the statements

```
SOLVE VGATE=1.0
SOLVE VGATE=2.0
```

first solves a single bias point with 1.0V and then 2.0V on the gate electrode. One very important rule in ATLAS is that when the voltage on any electrode is **not** specified in a given SOLVE statement, the value from the last SOLVE statement is assumed.

In the following case, the second solution is for a drain voltage of 1.0V and a gate voltage of 2.0V.

```
SOLVE VGATE=2.0
SOLVE VDRAIN=1.0
```

When the voltage on a particular electrode is never defined on any SOLVE statement, that voltage is zero. Therefore, it is not necessary to explicitly state the voltage on all electrodes on all SOLVE statements. In a MOSFET for example, if VSUBSTRATE is not specified, then Vbs defaults to zero.

Sweeping The Bias

For most applications, a sweep of one or more electrodes is usually required. The basic DC stepping is inconvenient and a ramped bias should be used. To ramp the base voltage from 0.0V to 1.0V with 0.05V steps with a fixed collector voltage of 2.0V, the following syntax is used:

```
SOLVE VCOLLECTOR=2.0
SOLVE VBASE=0.0 VSTEP=0.05 VFINAL=1.0 NAME=base
```

The NAME parameter is required and the electrode name is case-sensitive. It is up to the user to ensure the initial voltage, VSTEP and VFINAL are consistent. A badly specified ramp from zero to 1.5V in 0.2V steps would finish at 1.4V or 1.6V.

Generating Families Of Curves

Many applications such as MOSFET Id/Vds and bipolar Ic/Vce simulations require that a family of curves is produced. This is done by obtaining solutions at each of the stepped bias points first, and then solving over the swept bias variable at each stepped point. For example, in MOSFET Id/Vds curves, solutions for each Vgs value are obtained with Vds=0.0V. The output from these solutions are saved in ATLAS solution files. Then, in turn for each gate bias, the solution file is loaded and the ramp of drain voltage performed.

The family of curves for three 1V gate steps and a 3.3V drain sweep would be implemented in ATLAS as follows:

```
SOLVE VGATE=1.0  OUTF=solve_vgate1
SOLVE VGATE=2.0  OUTF=solve_vgate2
SOLVE VGATE=3.0  OUTF=solve_vgate3

LOAD INFILE=solve_vgate1
LOG OUTFILE=mos_drain_sweep1
SOLVE NAME=drain VDRAIN=0 VFINAL=3.3 VSTEP=0.3

LOAD INFILE=solve_vgate2
LOG OUTFILE=mos_drain_sweep2
SOLVE NAME=drain VDRAIN=0 VFINAL=3.3 VSTEP=0.3

LOAD INFILE=solve_vgate3
LOG OUTFILE=mos_drain_sweep3
SOLVE NAME=drain VDRAIN=0 VFINAL=3.3 VSTEP=0.3
```

The LOG statements are used to save the Id/Vds curve from each gate voltage to separate files. It is recommended to save the data in this manner rather than to a single LOG file. LOG files are described later in this chapter.

The Importance Of The Initial Guess

To obtain convergence for the equations used it is necessary to supply a good initial guess for the variables to be evaluated at each bias point. The ATLAS solver uses this initial guess and iterates to a converged solution. For isothermal drift diffusion simulations, the variables are the potential and the two carrier concentrations. Provided a reasonable grid is used, almost all convergence problems in ATLAS are caused by a poor initial guess to the solution.

During a bias ramp the initial guess for any bias point is provided by a projection of the two previous results. Problems tend to appear near the beginning of the ramp when two previous results are not available. If one previous bias is available it is used alone.

This explains why although the following two examples eventually produce the same result, the first will likely have far more convergence problems than the second.

```
1.  SOLVE VGATE=1.0 VDRAIN=1.0 VSUBSTRATE=1.0
2.  SOLVE VGATE=1.0
    SOLVE VSUBSTRATE=-1.0
    SOLVE VDRAIN=1.0
```

In the first case, one solution is obtained with all specified electrodes at 1.0V. In the second case, the solution with only the gate voltage at 1.0V is performed first. All other electrodes are at zero bias. Next, with the gate at 1.0V, the substrate potential is raised to -1.0V and another solution is obtained. Finally, with the substrate and the gate biased, the drain potential is added and the system solved again. The advantage of this method over the first case is that the small incremental changes in voltage allow for better initial guesses at each step.

Generally the projection method for the initial guess gives good results when the I-V curve is linear. However, it may encounter problems if the IV curve is highly non-linear or if the device operating

mode is changing. Typically this might occur around the threshold or breakdown voltages. At these biases smaller voltage steps are required to obtain convergence. As will be described, ATLAS contains features such as the TRAP parameter and the curve tracer to automatically cut the voltage steps in these highly non-linear area.

Numerical methods are described earlier in this chapter. In many cases these methods are designed to overcome the problems associated with the initial guess. This is particularly important in simulations involving more than the three drift diffusion variables. In general coupled solutions require a good initial guess, whereas de-coupled solutions can converge with a poor initial guess.

The Initial Solution

When no previous solutions exist, the initial guess for potential and carrier concentrations must be made from the doping profile. This is why the initial solution performed must be the zero bias (or thermal equilibrium) case. This is specified by the

```
SOLVE INIT
```

statement. However, if this syntax is not specified, ATLAS will automatically evaluate an initial solution before the first SOLVE statement. In order to aid convergence of this initial guess, it is performed in the zero carrier mode solving only for potential.

The First And Second Non-Zero Bias Solutions

From experience with ATLAS, it is found that the first and second non-zero bias solutions are the most difficult in which to obtain good convergence. The reason is clear. Once these two solutions are obtained the projection algorithm for the initial guess is available and solutions should all have a good initial guess.

However, these first two solutions must use the result of the initial solution as the basis of their initial guess. Since the initial solution is at zero bias, it provides a very poor initial guess.

The practical result of this is that the first and second non-zero bias solutions should have very small voltage steps. In the following example the first case will likely converge whereas the second case may not.

```
1.  SOLVE INIT
    SOLVE VDRAIN=0.1
    SOLVE VDRAIN=0.2
    SOLVE VDRAIN=2.0
2.  SOLVE INIT
    SOLVE VDRAIN=2.0
```

The Trap Parameter

Although ATLAS provides several features to overcome a poor initial guess and other convergence problems, it is important to understand the role of the initial guess in obtaining each solution. The simplest and most effective is the syntax:

```
METHOD TRAP
```

This is enabled by default. Its effect is to reduce the bias step if convergence problems are detected. Consider the example from the previous section:

```
SOLVE INIT
SOLVE VDRAIN=2.0
```

If the second SOLVE statement does not converge, TRAP will automatically cut the bias step in half and try to obtain a solution for Vd = 1.0V. If this solution does not converge the bias stepwill be halved again to solve for Vd = 0.5V. This procedure is repeated up to a maximum number of tries set by the METHOD parameter MAXTRAPS. Once convergence is obtained the bias steps are increased again to solve up to 2.0V. The default for MAXTRAPS is four and it is not recommended to increase it since changing the syntax to use smaller bias steps is generally much faster.

This trap facility is very useful during bias ramps in overcoming convergence difficulties around transition points such as the threshold voltage. Consider the following syntax used to extract a MOSFET Id/Vgs curve:

```
SOLVE VGATE=0.0 VSTEP=0.2 VFINAL=5.0 NAME=gate
```

Assume the threshold voltage for the device being simulated is 0.7V and that ATLAS has solved for the gate voltages up to 0.6V. The next solution, at 0.8V, might not converge at first. This is because the initial guess was formed from the two sub-threshold results at Vgs=0.4V and 0.6V and the solution has now become non-linear. The trap facility will detect the problems in the 0.8V solution and cut the bias step in half to 0.7V and try again. This will probably converge. The solution for 0.8V will then be performed, and the bias ramp will continue with 0.2V steps.

Small-Signal AC Solutions

Specifying AC simulations is a simple extension of the DC solution syntax. AC small signal analysis is performed as a post-processing operation to a DC solution. Two common types of AC simulation in ATLAS are outlined here. The results of AC simulations are the conductance and capacitance between each pair of electrodes. Pointers on interpreting these results are given later in this chapter.

Single Frequency AC Solution During A DC Ramp

The minimum syntax to set an AC signal on an existing DC ramp is just the AC flag and the setting of the small signal frequency.

```
SOLVE VBASE=0.0 VSTEP=0.05 VFINAL=1.0 NAME=base AC FREQ=1.0e6
```

Other AC syntax for setting the signal magnitude and other parameters are generally not needed as the defaults suffice. One exception is in 1D MOS capacitor simulations. To obtain convergence in the inversion/deep depletion region the parameter DIRECT should be added to access a more robust solution method.

Ramped Frequency At A Single Bias

For some applications such as determining bipolar gain versus frequency it is necessary to ramp the frequency of the simulation. This is done using the following syntax:

```
1.  SOLVE VBASE=0.7 AC FREQ=1e9 FSTEP=1e9 NFSTEPS=10
2.  SOLVE VBASE=0.7 AC FREQ=1e6 FSTEP=2 MULT.F NFSTEPS=10
```

The first case ramps the frequency from 1GHz to 11GHz in 1GHz steps. A linear ramp of frequency is used and FSTEP is in Hertz. In the second example a larger frequency range is desired and so a geometrical step of frequency is used. The MULT.F parameter is used to specify that FSTEP is a unitless multiplier for the frequency. This doubles the frequency in successive steps from 1Mhz to 1.024GHz.

The syntax described here for ramping the frequency of the AC signal can potentially be combined with that for ramping the bias. The frequency ramps are done as inner loops to the DC ramping.

```
SOLVE VBASE=0.0 VSTEP=0.05 VFINAL=1.0 NAME=base AC FREQ=1.0e6 \
      FSTEP=2 MULT.F NFSTEPS=10
```

Transient Solutions

Transient solutions can be obtained for piecewise-linear, exponential and sinusoidal bias functions. Transient solutions are used when a time dependent test or response is required. To obtain transient solutions for a linear ramp, the TSTART, TSTOP, TSTEP and RAMPTIME parameters should be specified. The TSTART parameter specifies the time that the linear ramp should start. The RAMPTIME specifies the time that the linear ramp should obtain its final value. TSTOP specifies the time that solutions will stop. TSTEP specifies the initial step size. Subsequent time steps are calculated automatically by ATLAS. For example, the statement

```
SOLVE VGATE=1.0 RAMPTIME=1E-9 TSTOP=10e-9 TSTEP=1e-11
```

specifies that the voltage on the gate electrode will be ramped in the time domain, from its present value to 1.0V over a period of 1 nanoseconds. Time domain solutions are obtained for an additional 9 nanoseconds. An initial time step of 10 picoseconds is specified. It is important to remember that if subsequent transient solutions are specified, the time is not reset to zero.

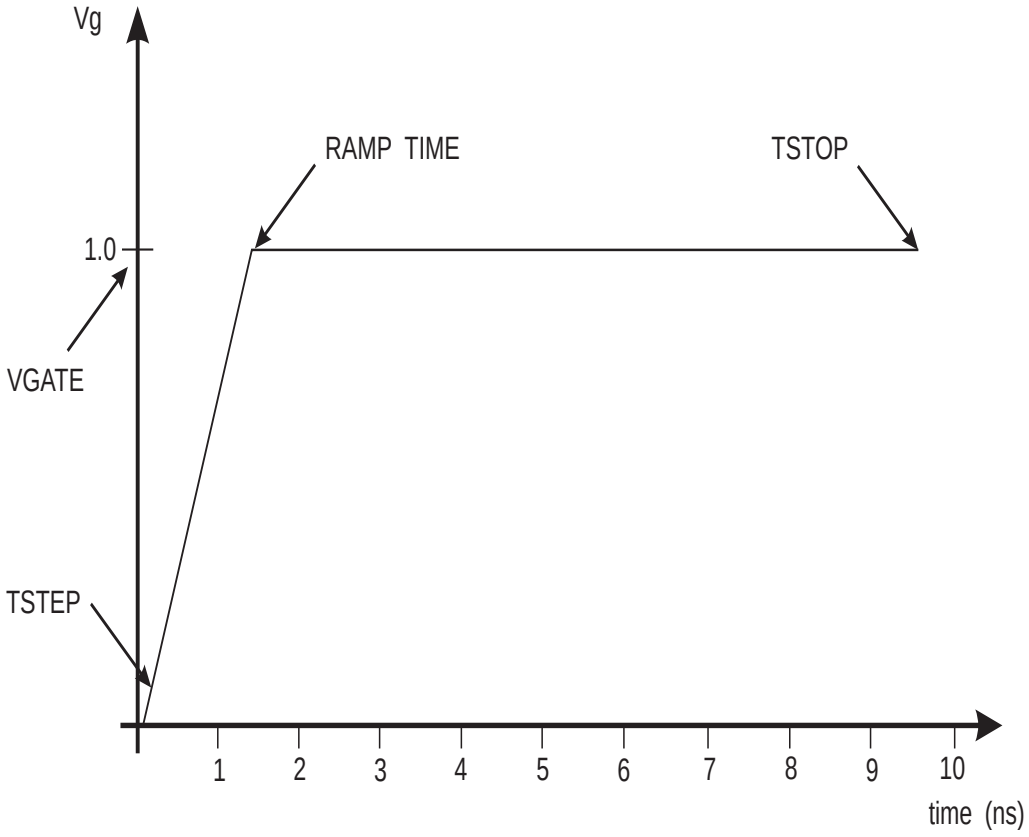

Figure 2-7: Diagram showing syntax of Transient Voltage Ramp in ATLAS

Advanced Solution Techniques

Obtaining Solutions Around The Breakdown Voltage

Obtaining solutions around the breakdown voltage can be difficult using the standard ATLAS approach. It requires special care in the choice of voltage steps and also in interpreting the results. The curve tracer described later is the most effective method in many cases.

A MOSFET breakdown simulation might be performed using this standard syntax for ramping the drain bias. Note the setting of CLIMIT as required for breakdown simulations when the pre-breakdown leakage is low.

```
IMPACT SELB
METHOD CLIMIT=1e-4
SOLVE VDRAIN=1.0 VSTEP=1.0 VFINAL=20.0 NAME=drain
```

If the breakdown were 11.5V then convergence problems will be expected for biases higher than 11.0V using this syntax. Although technology dependent, it is common for the breakdown curve to be flat up to a voltage very close to breakdown and then almost vertical. The current changes by orders of magnitude for very small bias increments.

This produces some problems for ATLAS using the syntax described above. First, if the breakdown occurs at 11.5V, there are no solutions for voltages greater than this value. ATLAS is trying to ramp to 20.0V so it is likely that ATLAS will fail to converge at some point. This is usually not a problem since by that point the breakdown voltage and curve have been obtained.

Above 11V, bias step reduction will take place due to the TRAP parameter. ATLAS will continually try to increase the drain voltage above 11.5V and those points will fail to converge. However it will solve points asymptotically approaching Vds=11.5V until the limit set by the MAXTRAPS parameter is reached. If the default of 4 traps is used it is clear that the minimum allowed voltage step is 1.0x(0.5)4 or 0.004V. This is normally enough accuracy for determining the breakdown point.

However the simulation might not allow the current to reach a sufficiently high level before MAXTRAPS is needed. Typically in device simulation, the breakdown point is determined once the current is seen to increase above the flat pre-breakdown leakage value by two orders of magnitude in a small voltage increment. If users do wish to trace the full breakdown curve up to high current values more advanced techniques than the simple voltage ramp must be used. Two of them are described below: curve tracer and current boundary conditions. The expense of these methods might be extra CPU time.

Using Current Boundary Conditions

In all of the examples considered in the basic description of the SOLVE statement it was assumed that voltages were being forced and currents were being measured. ATLAS also supports the reverse case through current boundary conditions. The current through the electrode is specified in the SOLVE statement and the voltage at the contact is calculated. Current boundary conditions are set using the CONTACT statement as described earlier in this chapter.

The syntax of the SOLVE statement is altered once current boundary conditions are specified.

```
SOLVE IBASE=1e-6
```

The syntax above specifies a single solution at a given current.

```
SOLVE IBASE=1e-6 ISTEP=1e-6 IFINAL=5e-6 NAME=base
```

This sets a current ramp similar in syntax to the voltage ramp described earlier.

```
SOLVE IBASE=1e-10 ISTEP=10 IMULT IFINAL=1e-6 NAME=base
```

This is similar to the previous case, but the IMULT parameter is used to specify that ISTEP should be used as a multiplier for the current rather than a linear addition. This is typical for ramps of current since linear ramps are inconvenient when several orders of magnitude in current may need to be covered.

Important points to remember about current boundary conditions are that the problems of initial guess are more acute when very small (noise level) currents are used. Often it is best to ramp the voltage until the current is above 1pA/μm and then switch to current forcing.

When interpreting the results, it is important to remember the calculated voltage on the electrode with current boundary conditions is stored as the 'internal bias' (eg. base int.bias in TONYPLOT or vint."base" in DECKBUILD's extract syntax).

The Compliance Parameter

Compliance is a parameter used to limit the current or voltage through or on an electrode during a simulation. An electrode compliance can be set and after it is reached, the bias sweep will stop. This is analogous to parametric device testing when we stop a device from being over stressed or destroyed. The compliance refers to the maximum resultant current or voltage present after a solution is obtained. If an electrode voltage is set, then the compliance refers to the electrode current. If current boundary conditions are used, then a voltage compliance can be set.

The statements

```
SOLVE VGATE=1.0
SOLVE NAME=drain VDRAIN=0 VFINAL=2 VSTEP=0.2 COMPL=1E-6 CNAME=drain
```

first solves for 1V on the gate and then ramps the drain voltage towards 2V in 0.2V steps. If 1μA/μm of drain current is reached before Vd = 2V, the simulation will stop. Thus, as in parametric testing, a particular level can be defined and the simulation can be set to solve up to that point and no further. Once the compliance limit is reached ATLAS simulates the next statement line in the command file.

The Curvetrace Capability

The automatic curve tracing algorithm can be invoked to enable ATLAS to trace out complex IV curves. The algorithm can automatically switch from voltage to current boundary conditions and vice versa. A single SOLVE statement may be used to trace out complex IV curves such as breakdown curves and CMOS latch-up including the snapback region and second breakdown. The algorithm is based upon a dynamic load line approach.

For example, typical curvetrace and solve statements to trace out an IV curve for the breakdown of a diode would look like:

```
CURVETRACE CONTR.NAME=cathode STEP.INIT=0.5 NEXT.RATIO=1.2 \
           MINCUR=1e-12 END.VAL=1e-3 CURR.CONT
SOLVE CURVETRACE
```

The name of the electrode which is to be ramped is specified using CONTR.NAME. STEP.INIT specifies the initial voltage step. NEXT.RATIO specifies the factor used to increase the voltage step in areas on the IV curve away from turning points. MINCUR may be used to set a small current value above which the dynamic load line algorithm is activated. Below the MINCUR level, the STEP.INIT and NEXT.RATIO are used to determine the next solution bias. END.VAL is used to stop the tracing if the voltage or current of the ramped electrode equals or exceeds END.VAL. Either VOLT.CONT or CURR.CONT is used to specify whether END.VAL is a voltage or current value.

When plotting the log file created by the curve trace statement in TONYPLOT, it is necessary to select the internal bias, labelled int.bias, for the ramped electrode instead of the plotting the applied bias, which is labelled *Voltage*.

Using DeckBuild To Specify SOLVE Statements

The DECKBUILD Solve menu can be used generate SOLVE statements. The menu has a spreadsheet style entry. The **Solve** menu may be accessed by selecting the **Command/Solutions/Solve...** button in DECKBUILD. To define a test, the right mouse button is depressed in the worksheet and the first option in the menu "Add new row" should be selected. This will add a new row to the worksheet. This procedure should be repeated once per electrode in your device structure. The entry for each cell may then be edited to construct a SOLVE statement. Some cells require the selection using a pop menu or the entry of numerical values. The electrode name is specified in the first cell. It is edited using a popup menu that is accessed by pressing the right menu on the cell. The second cell specifies whether the electrode will be a voltage (V), current (I) or charge (Q) controlled. The third cell specifies whether the solve statement is to be a single DC solve (CONST), a swept DC variable (VAR 1), a stepped DC variable (VAR2), or a transient solution (PULSE). The remaining cells specify the parameter values that are required for the type of solution desired.

The pop-up window to specify the solution file names are accessed through the **Props...** button. Several solve statements may be constructed to create solve sequences which define a test. This test may be saved in a file and read in using the **Save...** and **Load...** buttons. A powerful features of the DECKBUILD SOLVE menu is the ability to generate a family of curves using the sweep and step variables.

Interpreting The Results

As indicated in Figure 2-1, ATLAS produces three different types of output. To recap, these are:

- RUN-TIME OUTPUT

  This stores the run-time messages produced by ATLAS. These messages typically include important values extracted from the simulation. All error messages go to the run-time output. If a simulation fails, it is extremely important to check the run-time output for error and warning messages.
- LOG FILES

  Store the DC, small-signal AC and transient terminal characteristics for a sequence of SOLVE statements. They are loaded into TONYPLOT to visualize the device behavior.
- SOLUTION FILES

  These store physical quantities of the structure at each grid node for a single bias point. These can be viewed in TONYPLOT to see the internal distributions of parameters (eg. potential, electric field). They can also be loaded into other ATLAS runs to re-initialize ATLAS at non-zero biases.

Run-Time Output

Run time output is provided in the bottom of the DECKBUILD window. If run as a batch job, the run-time output can be stored to a file.

Errors occuring in the run-time output will be displayed in this window. Note that not all errors will be fatal (as deckbuild tries to interprete the users file and continue). This may cause a statement to be ignored, leading to unexpected results. It is recommended that the user check the run-time output of any newly created input file, the first time it is run, to intercept any errors.

If the user specifies the `PRINT` option within the `MODELS` statement, details of material parameters, constants and mobility models will be specified at the start of the run time output. This is a useful way of checking what has been specified and which mobility parameters apply to which regions. It is recommended that the user always specifies `MODELS PRINT` in input files.

During `SOLVE` statements the error numbers of each equation at each iteration are displayed. This is a change from the previous ATLAS version. It is is not vital for users to understand the iteration information but it may provide important insights in the case of convergence problems

The output can be interpreted as follows:

```
proj      psi     n      p       psi      n          p
direct    x       x      x       rhs      rhs        rhs

  i    j   m  -5.00*   -5.00*   -5.00*   -26.0*   -17.3*   -17.3*
----------------------------
1        N   -1.932   -2.934   -1.932     -25.2    -10.19   -9.876
2        N   -4.741   -5.64*   -4.267     -28.8*   -16.67   -15.47
3        A   -11.3*   -11.7*   -9.63*     -28.8*   -16.67   -18.0*
```

Electrode	Va(V)	Jn(A/um)	Jp(A/um)	Jc(A/um)	Jt(A/um)
gate	0.000e+00	-0.000e+00	-0.000e+00	0.000e+00	0.000e+00
source	0.000e+00	-3.138e-13	-1.089e-35	-3.138e-13	-3.138e-13
drain	1.000e-01	3.139e-13	1.076e-23	3.139e-13	3.139e-13
substrate	0.000e+00	-6.469e-19	-8.853e-17	-8.918e-17	-8.918e-17

The top left value 'proj' indicates the initial guess methodology used. Here it is the default projection method. Alternatives are previous, local or init. The second value 'direct' indicates the solver type. This will either be direct or iterative.

The first three column headings i, j, m indicate the iteration numbers of the solution and the solution method. i indicates the outer loop iteration number for decoupled solutions. j indicates the inner loop number. m indicates the solution method by a single letter these are:

```
G = gummel
B = block
N = newton
A = newton with autonr
S = coupled Poisson-Schrodinger solution
```

The remaining column headings indicate which column lists the XNORM and RHSNORM errors for the equations being solved. See the Numerical Methods Chapter for a full description of these errors. The values printed in each error column under the hashed line are the logarithm to base 10 of the error. Earlier PISCES versions would print the floating point value. The values printed above the hashed line in each column are the tolerances used.

When the '*' symbol appears as the least significant digit in the number it means this error measure has met its tolerance.

After convergence is achieved ATLAS lists the results by electrode. The column *Va* lists the voltage at the contact surface. This will differ from the applied voltage if external resistors or the curvetracer are used. All relevant current components are listed. Here only electron, hole, conduction and total currents are given. In other modes of simulation these columns may differ.

The output of AC analysis, MIXEDMODE and 3D simulations differ from this standard.

ATLAS may produce a very large amount of run-time output for complex simulations. Run-Time output may be saved to a file as shown in the "Modes of Operation" section of this chapter.

Log Files

Log files store the terminal characteristics calculated by ATLAS. These are current and voltages for each electrode in DC simulations. In transient simulations the time is stored and in AC simulations the small signal frequency and the conductances and capacitances are saved. The statement:

```
LOG OUTF=<FILENAME>
```

is used to open a log file. Terminal characteristics from all SOLVE statements after the LOG statement are then saved to this file along with any results from the PROBE statement.

The only way to stop the terminal characteristics being saved to this file is to use another LOG statement with either a different log filename or the parameter OFF.

Typically a separate log file should be used for each bias sweep. For example, separate log files are used for each gate bias in a MOS Id/Vds simulation or each base current in a bipolar Ic/Vce simulation. These files are then overlaid in TONYPLOT.

Log files contain only the terminal characteristics. They are typically viewed in TONYPLOT. Parameter extraction on data in log files can be done in DECKBUILD. Log files cannot be loaded into ATLAS to re-initialize the simulation.

Units Of Currents In Log files

In general the units of current written into the log file and hence seen in TONYPLOT is Amperes per micron. This is because ATLAS is a two-dimensional simulator. It sets the third dimension (or z-direction) to be one micron. Thus if you compare ATLAS 2D simulation results for a MOSFET versus the measured data from a MOSFET of width 20 micron, you need to multiply the current in the log file by 20

There are four exceptions:

- In the 3D modules of ATLAS, the width is defined in the 3D structure and so the units of the current are Amperes.
- In MIXEDMODE, the width of devices is set by the user so again the current is in Amperes.
- When cylindrical coordinates are used, the current written to the log file is integrated through the cylinder and is also in Amperes.
- When the WIDTH parameter on the MESH statement is used then the current is scaled by this factor and is in Amperes

Similar rules apply for the capacitance and conductance produced by AC simulations. These are usually in 1/(ohms.microns) and Farads/micron respectively..

Parameter Extraction In DeckBuild

The EXTRACT command is provided within the DECKBUILD environment. It allows you to extract device parameters. The command has a flexible syntax that allows you to construct very specific extract routines. EXTRACT operates on the previous solved curve or structure file. By default EXTRACT uses the currently open log file. To override this default the name of a file to be used by extract can be supplied before the extraction routine in the following way:

```
EXTRACT INIT INF="<filename>"
```

A typical example of the use of extract is the extraction of the threshold voltage of an MOS transistor. In the following example, the threshold voltage is extracted by calculating the maximum slope of the I_d / V_g curve, finding the intercept with the x-axis and then subtracting half of the applied drain bias:

```
EXTRACT NAME="nvt" XINTERCEPT(MAXSLOPE(CURVE (V."GATE", (I."DRAIN"))) \
-(AVE(V."DRAIN"))/2.0)
```

The results of the extraction will be displayed in the run-time output and will by default also be stored in the file 'results.final'. You can store the results in a different file by using the the following option at the end of extract command:

```
EXTRACT....DATAFILE="<filename>"
```

Cut off frequency and forward current gain are of particular use as output parameters. These functions can be defined as follows:

```
# MAXIMUM CUTOFF FREQUENCY
EXTRACT NAME="FT_MAX" MAX(G."COLLECTOR""BASE"/(6.28*C."BASE""BASE"))

#FORWARD CURRENT GAIN
EXTRACT NAME="PEAK GAIN" MAX(I."COLLECTOR"/ I."BASE")
```

Note: Over 300 examples are supplied with ATLAS to provide many practical examples of the use of the EXTRACT statment

---

EXTRACT has two additional important functions:

1. It provides data for the VWF database - i.e. to store device parameters to the VWF database, they must be evaluated using EXTRACT.
2. When using the DECKBUILD Optimizer to tune parameters, EXTRACT statements must be used as the optimization targets.

Functions In TonyPlot

The **Functions** Menu in TONYPLOT allows you to specify and plot functions of the terminal characteristics in the **Graph Function** text fields. For example transconductance can be calculated using the following function:

```
dydx (drain current, gate voltage)
```

Current gain can be evaluated as:

```
collector current / base current
```

When creating functions, the key to correct syntax is that the name for any variable in a function is the same as that in the **Y Quantities** list on the **Display** menu.

AC Parameter Extraction

Basic analysis of the capacitance and conductance matrix produced by ATLAS can be done using DECKBUILD or TONYPLOT. The capacitance between gate and drain will be labeled as `C gate>drain` in Tonyplot or `c."gate""drain"` in DECKBUILD'S EXTRACT.

The total capacitance on any electrode is defined as C electrode>electrode. Thus the magnitude of C gate>gate is total gate capacitance.

The LOG statement also includes options for small-signal, two-port RF analysis including s-parameter extraction. The solutions are saved into the log file and also in the run-time output. The list of options for RF analysis is:

```
s.param, y.param, h.param, z.param, abcd.param gains
```

Terminal impedance and parasitics are accounted for by adding any of the following:

```
impedance=<val>, rin=<val>, rout=<val>, rcommon=<val> or
rground=<val>, lin=<val>, lout=<val>,
lcommon=<val> or lground=<val>, width=<val>
```

The width defaults to 1μm and impedance defaults to 50Ω . All parasitics default to zero.

The Stern stability factor k is calculated along with current gain (h21), GUmax, and GTmax when the GAINS option is added to the LOG statement.

The run-time output for AC analysis has been modified to only list the analysis frequency, and electrode conductance/capacitance values. If one of the two-port options is added to the LOG statement (such as S.PARAM),the two-port parameters are also included in the run-time output.

The UTMOST statement AC parameter conversion utilities have been discontinued.

UTMOST Interface

ATLAS log files can be read directly into batch mode UTMOST. The following commands in UTMOST are used to read in a set of IV curves stored in separate log files.

```
INIT INF=<filename> MASTER
INIT INF=<filename> MASTER APPEND
```

Use of the older UTMOST statement in ATLAS is no longer recommended for interfacing to UTMOST.

Solution Files

Solution files or structure files provide a 'snap shot' of the device at a particular bias point (DC solution or transient solution point). This gives the user the ability to view any evaluated quantity within the device structure in question, from doping profiles and band parameters, to electron concentrations and electric fields. These files should be plotted using TONYPLOT.

The syntax used to generate these files is of two forms:

```
(1)   SAVE OUTFILE=<filename>
```

Here a file named <filename> will be saved with data from the previously solved bias point.

```
(2)   SOLVE .... OUTFILE=<filename>.sta MASTER [ONEFILEONLY]
```

In this case, a structure file will be saved at each bias point solved in the solve statement. The last letter of the file name will be automatically incremented alphabetically *.sta, *.stb, *.stc... and so on. If the solution for the last bias point only is required, the parameter `onefileonly` can be added to the command. The file <filename>.sta will be over written at each solution point.

Structure files can be very large (1 - 2 MB) depending on the mesh density and quantities saved. It is recommended that unwanted structure files are deleted.

If many solution files have been written from a long simulation it is often confusing to find out which solution file belongs to which bias point or transient time. A utility has been written by Silvaco to allow users to see the biasing conditions of each file. To obtain this utility check the Silvaco internet web site or contact your local Silvaco support engineer.

The solution files should be plotted in TONYPLOT. Using TONYPLOT, it is possible to create 2D contour plots and 1D cutlines. To find out the bias of any solution file in TONYPLOT, select the plot and press 'b' on the keyboard.

Interpreting Contour Plots

Most quantities saved in the solution files are never evaluated at the node points during solutions rather at the center of the side of each triangle in the mesh. Values of quantities at each node are derived from averaging the values from the sides of triangles connected to that node. The weighting method used to do the averaging can be selected by the user with options on the OUTPUT statement. It is possible that for some meshes smoother contour plots can be obtained by choosing a non-default averaging method.

When interpreting the contour plots it is important for users to remember that the solution file contains values only at each node point. The colour fills seen in Tonyplot are simply interpolations based on the node values. This may lead to occasional strange contour values. In these cases users should check the node values using the probe in TONYPLOT.

The primary solution variables (potential, carrier concentration, lattice temperature and carrier temperatures) are calculated on the nodes of the ATLAS mesh and hence are always correct in TONYPLOT. However since ATLAS does not use nodal values of quantities such as electric field and mobility the actual values being used in calculations cannot be determined from Tonyplot or the structure files. The PROBE statement allows users to directly probe values at given locations in the structure. This provides the most accurate way to determine the actual values used in ATLAS calculations.

Customizing Solution Files (OUTPUT Statement)

Several quantities are saved by default within a structure file. For example doping, electron concentration and potential. It is also possible to specify additional quantities with the output statement (such as conduction band potential), using the OUTPUT statement. This must precede the SAVE statement in question. For example, to save the conduction and valence band potentias the following command would be used at some point before the relevant SAVE.

```
OUTPUT CON.BAND VAL.BAND
```

Saving Quantities from the Structure at each Bias Point (PROBE statement)

Structure files provide all data from the structure at a single bias point. The Log files provide terminal characteristics for a set of bias points. To combine these and allow certain structural quantities to be saved at each bias point the PROBE statement is used.

The PROBE statement allows the user to specify quantities to be saved at given XY locations. There is also a facility to save the maximum or minimum of certain quantities. The value from the PROBE at each bias point in DC or timestep in transient mode is saved to the log file. The syntax:

```
PROBE NAME=mycarriers N.CONC X=1 Y=0.1
```

will save the electron concentration at (1,0.1) for each solution in the log file. When the log file is displayed in TONYPLOT the value will be labelled mycarriers. It will be possible to plot mycarriers versus terminal bias or current or other probled quantities.

Certain directionally dependent quantities such as electric field and mobility can be probed. In these cases a direction for the vector quantity must be specified using the DIR parameter.

The PROBE statement provides the only way to extract the actual values of quantities that are calculated along the sides of each triangle in ATLAS. The PROBE statement actually stored the triangle side value closest to the probed location while taking into account the direction for vector quantities. See the earlier section on "Interpreting Contour Plots".

Note: Specifying the probe location **exactly** at a material or region interface will often lead to erroneous results. It is best to very slightly offset the location of the probe inside the material or region of interest.

Re-initializing ATLAS at a Given Bias Point

Each solve statement will begin with the device biased at the previous value solved. If you wish to begin a solution at a previously solved bias point, you can re-load the structure file saved at that point. This is accomplished in the following manner:

```
LOAD INFILE=<filename> MASTER
```

Information about that solution point will be displayed in the output window.

This command is useful for solving a set of I/V curves. For example, to solve a family of Id / Vd (at various Vg) the gate can be ramped with zero drain bias, and a structure file saved at each desired value of Vg. These structure files can then be reloaded in turn while a Vd sweep is performed.

Note: An ATLAS input file **cannot** start with a LOAD statement. Prior to loading the structure file the user needs to ensure that the device mesh for the same structure has been loaded using the MESH statement. Also the same MODELS, MATERIAL and CONTACT settings are required as when the files was saved by ATLAS.

Technology Specific Issues in ATLAS

This chapter was designed to give an overview to the basic use of ATLAS without regard to the details required for a given technology. Requirements for ATLAS simulation vary considerably. The needs of the sub-micron MOS device engineer, the 1000V power device engineer and the III-V RF device engineer differ cannot all be covered in one chapter. Silvaco provides many references to individual technology problems using ATLAS. These are:

- a library of over 500 examples that can be accessed on-line from DECKBUILD. Users should look at these examples not only for their technology but also related ones. For example different aspects of high frequency analysis is covered in the MESFET and silicon bipolar example sections.
- the chapters on each individual ATLAS product in this manual. For example the S-PISCES chapter contains hints for EEPROM devices, the LUMINOUS chapter hints on photodetectors.
- the Simulation Standard newsletter distributed by Silvaco. To make sure you are on the mailing list contact your local Silvaco office
- the Silvaco World Wide Web Page: **www.silvaco.com**. This web site contains on-line versions of the articles in our newletter, on-line searchable index of the examples, links to other TCAD web sites and a section on solutions to known problems with all Silvaco programs**.**
- for more information about suggested technology specific strategies contact your local Silvaco support engineer.

Chapter 3: Physics

Basic Semiconductor Equations

Overview

Many years of research into device physics has resulted in a mathematical model of the operation of any semiconductor device. This model consists of a set of fundamental equations which link together the electrostatic potential and the carrier densities, within some simulation domain. These equations, which are solved inside any general purpose device simulator, have been derived from Maxwell's laws and consist of Poisson's equation, the continuity equations and the transport equations. Poisson's equation relates variations in electrostatic potential to local charge densities. The continuity equations describe the way that the electron and hole densities evolve as a result of transport processes, generation processes, and recombination processes.

This chapter shall describe the mathematical model implemented into ATLAS. However, it is important to note that a discretization of the equations is also performed so that they may be applied to the finite element grid used to represent the simulation domain.

Poisson's Equation

Poisson's equation relates the electrostatic potential to the space charge density:

$$\mathrm{div}(\varepsilon \nabla \psi) = -\rho \qquad 3\text{-}1$$

where ψ is the electrostatic potential, ε is the local permittivity, and ρ is the local space charge density. The reference potential can be defined in various ways. For ATLAS, this is always the intrinsic Fermi potential ψ_i which is defined in the next section. The local space charge density is the sum of contributions from all mobile and fixed charges, including electrons, holes, and ionized impurities.

The electric field is obtained from the gradient of the potential.

$$\vec{E} = \nabla \psi \qquad 3\text{-}2$$

Carrier Continuity Equations

The continuity equations for electrons and holes are defined by the equations;

$$\frac{\partial n}{\partial t} = \frac{1}{q}\mathrm{div}\vec{J_n} + G_n - R_n \qquad 3\text{-}3$$

$$\frac{\partial p}{\partial t} = \frac{1}{q}\mathrm{div}\vec{J_p} + G_p - R_p \qquad 3\text{-}4$$

where n and p are the electron and hole concentration, $\vec{J_n}$ and $\vec{J_p}$ are the electron and hole current densities, G_n and G_p are the generation rates for electrons and holes, R_n and R_p are the recombination rates for electrons and holes, and q is the magnitude of the charge on an electron.

By default ATLAS includes both Equations 3-3 and 3-4. However, in some circumstances it is sufficient to solve only one carrier continuity equation. The specification of which continuity equations are to be solved is performed on the METHOD statement by turning off any equation that is not to be solved. The syntax ^ELECTRONS or ^HOLES turns off the electron continuity equation and the hole continuity equation, respectively.

The Transport Equations

Equations 3-1, 3-3 and 3-4 provide the general framework for device simulation. However, further secondary equations are needed to specify particular physical models for $\vec{J}_n$, $\vec{J}_p$, G_n, R_n, G_p and R_p.

The current density equations, or charge transport models, are usually obtained by applying approximations and simplifications to the Boltzmann Transport Equation. These assumptions can result in a number of different transport models such as the drift-diffusion model, the energy balance model or the hydrodynamic model. The choice of the charge transport model will then have a major influence on the choice of generation and recombination models.

The simplest model of charge transport that is useful is the "drift-diffusion" model. This model has the attractive feature that it does not introduce any independent variables in addition to ψ, n and p. Until recently, the drift-diffusion model was adequate for nearly all devices that were technologically feasible. However, the drift-diffusion approximation becomes less accurate for smaller feature sizes. More advanced energy balance and hydrodynamic models are therefore becoming popular for simulating deep submicron devices. ATLAS supplies both drift-diffusion and advanced transport models.

The charge transport models and the models for generation and recombination in ATLAS make use of some concepts associated with carrier statistics. These concepts are summarized in a further section of this chapter that deals with the carrier statistics.

The Drift-Diffusion Transport Model

Derivations based upon the Boltzmann transport theory have shown that the current densities in the continuity equations may be approximated by a drift-diffusion mode [3.] In this case the current densities are expressed in terms of the quasi-Fermi levels ϕ_n and ϕ_p as:

$$\vec{J}_n = -q\mu_n n \nabla \phi_n \qquad 3\text{-}5$$

$$\vec{J}_p = -q\mu_p n \nabla \phi_p \qquad 3\text{-}6$$

where μ_n and μ_p are the electron and hole mobilities. The quasi-Fermi levels are then linked to the carrier concentrations and the potential through the two Boltzmann approximations:

$$n = n_{ie} \exp\left[\frac{q(\psi - \phi_n)}{kT_L}\right] \qquad 3\text{-}7$$

$$p = n_{ie} \exp\left[\frac{-q(\psi - \phi_p)}{kT_L}\right] \qquad 3\text{-}8$$

where n_{ie} is the effective intrinsic concentration and T_L is the lattice temperature. These two equations may then be re-written to define the quasi-Fermi potentials:

$$\phi_n = \psi - \frac{kT_L}{q}\ln\frac{n}{n_{ie}} \quad 3\text{-}9$$

$$\phi_p = \psi + \frac{kT_L}{q}\ln\frac{p}{n_{ie}} \quad 3\text{-}10$$

By substituting these equations into the current density expressions the following adapted current relationships are obtained;

$$\vec{J_n} = qD_n\nabla n - qn\mu_n\nabla\psi - \mu_n n(kT_L\nabla(\ln n_{ie})) \quad 3\text{-}11$$

$$\vec{J_p} = -qD_p\nabla p - qp\mu_p\nabla\psi + \mu_p p(kT_L\nabla(\ln n_{ie})) \quad 3\text{-}12$$

The final term accounts for the gradient in the effective intrinsic carrier concentration, which takes account of bandgap narrowing effects. Effective electric fields are normally defined whereby,

$$\vec{E}_n = -\nabla\left(\psi + \frac{kT_L}{q}\ln n_{ie}\right) \quad 3\text{-}13$$

$$\vec{E}_p = -\nabla\left(\psi - \frac{kT_L}{q}\ln n_{ie}\right) \quad 3\text{-}14$$

Which then allows the more conventional formulation of drift-diffusion equations to be written:

$$\vec{J_n} = qn\mu_n\vec{E_n} + qD_n\nabla n \quad 3\text{-}15$$

$$\vec{J_p} = qp\mu_p\vec{E_p} - qD_p\nabla p \quad 3\text{-}16$$

It should be noted that this derivation of the drift-diffusion model has tacitly assumed that the Einstein relationship holds. In the case of Boltzmann statistics this correspondds to:

$$D_n = \frac{kT_L}{q}\mu_n \quad 3\text{-}17$$

$$D_p = \frac{kT_L}{q}\mu_p \quad 3\text{-}18$$

If Fermi-Dirac statistics are assumed for electrons, these become:

$$D_n = \frac{\left(\frac{kT_L}{q}\mu_n\right)F_{1/2}\left\{\frac{1}{kT_L}[\varepsilon_{Fn} - \varepsilon_C]\right\}}{F_{-1/2}\left\{\frac{1}{kT_L}[\varepsilon_{Fn} - \varepsilon_C]\right\}} \quad 3\text{-}19$$

where F_α is the Fermi-Dirac integral of order α and ε_{Fn} is given by $-q\phi_n$.

Note: The effects resulting from bandgap narrowing and their implementation into ATLAS are described in detail in a later section of this chapter.

Energy Balance Transport Model

A higher order solution to the general Boltzmann Transport Equation consists of an additional coupling of the current density to the carrier temperature, or energy. The current density expressions from the drift-diffusion model are modified to include this additional physical relationship. The electron current and energy flux densities are then expressed as:

$$\vec{J_n} = qD_n\nabla_n - \mu_n n\nabla\psi + qnD_n^T\nabla T_n \quad 3\text{-}20$$

$$\vec{S_n} = -K_n\nabla T_n - \left(\frac{k\delta_n}{q}\right)\vec{J}_n T_n \quad 3\text{-}21$$

$$\vec{J_p} = qD_p\nabla_p - \mu_p p\nabla\psi + qpD_p^T\nabla T_p \quad 3\text{-}22$$

$$\vec{S_p} = -K_p\nabla T_p - \left(\frac{k\delta_p}{q}\right)\vec{J}_p T_p \quad 3\text{-}23$$

Where T_n and T_p represent the electron and hole carrier temperatures and S_n and S_p are the flux of enery (or heat) from the carrier to the lattice. The energy balance transport model includes a number of very complex relationships and therefore a later section of this chapter has been devoted to this model.

Displacement Current Equation

For time domain simulation, the displacement current is calculated and included in the structure, log file and the run time output. The expression for displacement current is given as:

$$\overrightarrow{J_{dis}} = \varepsilon\left(\frac{\partial\overline{E}}{\partial t}\right) \quad . \quad 3\text{-}24$$

Basic Theory of Carrier Statistics

Fermi-Dirac and Boltzmann Statistics

Electrons in thermal equilibrium at temperature T_L with a semiconductor lattice obey Fermi-Dirac statistics. That is the probability f(ε) that an available electron state with energy ε is occupied by an electron is;

$$f(\varepsilon) = \frac{1}{1 + \exp\left(\frac{\varepsilon - E_F}{kT_L}\right)} \quad 3\text{-}25$$

where E_F is a spatially independent reference energy known as the Fermi level and k is Boltzmann's constant.

In the limit that $\varepsilon - E_F >> kT_L$ Equation 3-25 can be approximated as

$$f(\varepsilon) = \exp\left(\frac{E_F - \varepsilon}{kT_L}\right) \quad 3\text{-}26$$

Statistics based on the use of 3-26 are referred to as Boltzmann statistics. The use of Boltzmann statistics instead of Fermi-Dirac statistics makes subsequent calculations much simpler. The use of Boltzmann statistics is normally justified in semiconductor device theory, but Fermi-Dirac statistics are necessary to account for certain properties of very highly doped (degenerate) materials.

The remainder of this section outlines derivations and results for the simpler case of Boltzmann statistics which are the default in ATLAS. Users can specify that ATLAS is to use Fermi-Dirac statistics by specifying the parameter `FERMIDIRAC` on the `MODEL` statement.

Effective Density of States

Integrating the Fermi-Dirac statistics over a parabolic density of states in the conduction and valence bands, whose energy minimum is located at energies E_C and E_V respectively, yields the following expressions for the electron and hole concentrations;

$$n = N_C \exp\left(\frac{E_F - E_C}{kT_L}\right) \quad 3\text{-}27$$

$$p = N_V \exp\left(\frac{E_v - E_F}{kT_L}\right) \quad 3\text{-}28$$

N_C and N_V are referred to as the effective density of states for electrons and holes, and are given by;

$$N_C(T_L) = 2\left(\frac{2\pi\, m_e^* kT_L}{h_2}\right)^{\frac{3}{2}} = \left(\frac{T_L}{300}\right)^{\frac{3}{2}} \mathbf{NC300} \quad 3\text{-}29$$

$$N_V(T_L) = 2\left(\frac{2\pi m_h^*\, kT_L}{h^2}\right)^{\frac{3}{2}} = \left(\frac{T_L}{300}\right)^{\frac{3}{2}} \mathbf{NV300} \quad 3\text{-}30$$

where **NC300** and **NV300** are user definable on the MATERIAL statement as shown in Table 3-1.

Table 3-1. User Definable Parameters for the Density of States

Statement	Parameter	Default	Units
MATERIAL	NC300	2.8e19	cm^{-3}
MATERIAL	NV300	2.8e19	cm^{-3}

Intrinsic Carrier Concentration

Multiplying Equation 3-27 and 3-28 yields

$$np = n_{ie}^2 \qquad 3\text{-}31$$

where n_{ie} is the intrinsic carrier concentration and is given by

$$n_{ie} = \sqrt{N_C N_V}\exp\left(\frac{-E_g}{2kT_L}\right) \qquad 3\text{-}32$$

and $E_g = E_C - E_V$ is the band-gap energy.

For intrinsic (undoped) material, $p = n$. Equating 3-27 and 3-28 and solving for E_F yields

$$E_F = E_i = -q\psi_i = \frac{E_C + E_V}{2} = \frac{kT_L}{2}ln\left(\frac{N_v}{N_c}\right) \qquad 3\text{-}33$$

where E_i is the Fermi level for intrinsic doped silicon, and ψ_i is the intrinsic potential. Equation 3-33 defines the intrinsic potential under non-equilibrium conditions also. As indicated previously, for ATLAS the ψ used in Equation 3-1 is the intrinsic potential.

The electron and hole concentrations can be expressed in terms of the intrinsic carrier concentration as:

$$n = n_{ie}\exp\left[\frac{q(\psi - \phi_n)}{kT_L}\right] \qquad 3\text{-}34$$

$$p = n_{ie}\exp\left[\frac{-q(\psi - \phi_p)}{kT_L}\right] \qquad 3\text{-}35$$

where ψ is the intrinsic potential and ϕ is the potential corresponding to the Fermi level, i.e. $E_F = q\phi$.

The Energy Bandgap

The temperature dependence of the bandgap energy is modeled in ATLAS as follows:

$$E_g(T_L) = E_g(0) - \frac{\mathbf{EGALPHA}(T_L^2)}{T_L + \mathbf{EGBETA}} = \mathbf{EG300} + \mathbf{EGALPHA}\left[\frac{300^2}{300 + \mathbf{EGBETA}} - \frac{T_L^2}{T_L + \mathbf{EGBETA}}\right] \quad 3\text{-}36$$

The parameters EG300, EGALPHA and EGBETA can be specified by the user in the MATERIAL statement, as indicated in Table 3-2.

Table 3-2. User Specifiable Parameters for Equation 3-36

Statement	Parameter	Default	Units
MATERIAL	EG300	1.08	eV
MATERIAL	EGALPHA	4.73e-4	eV/K
MATERIAL	EGBETA	636	K
MATERIAL	NC300	2.8e19	cm^{-3}
MATERIAL	NV300	2.8e19	cm^{-3}

The default values are material dependent and may be found in Appendix B of this manual. Table 3-2 displays the defaults for Silicon only.

Bandgap Narrowing

In the presence of heavy doping, greater than $10^{18}cm^{-3}$, experimental work has shown that the pn product in silicon decomes doping dependent. As the doping level increases, a decrease in the bandgap separation occurs, where the conduction band is lowered by approximately the same amount as the valence band is raised. In ATLAS this is simulated by a spatially varying intrinsic concentration n_{ie} defined according to

$$n_{ie}^2 = n_i^2 \exp\left(\frac{\Delta E_g}{kT}\right) \quad 3\text{-}37$$

Bandgap narrowing effects in ATLAS are enabled by specifying the BGN parameter of the MODELS statement. These effects may be described by an analytic expression relating the variation in bandgap, ΔE_g, to the doping concentration, N. The expression used in ATLAS is from Slotboom [14]:

$$\Delta E_g = \mathbf{BGN.E}\left\{\ln\frac{N}{\mathbf{BGN.N}} + \left[\left(\ln\frac{N}{\mathbf{BGN.N}}\right)^2 + \mathbf{BGN.C}\right]^{\frac{1}{2}}\right\} \quad 3\text{-}38$$

Table 3-3. User Definable Parameters of Slotbooms Bandgap Narrowing Model			
Statement	**Parameter**	**Defaults**	**Units**
MATERIAL	BGN.E	6.92e-3	V
MATERIAL	BGN.N	1.3e17	cm^{-3}
MATERIAL	BGN.C	0.5	–

The parameters BGN.E, BGN.N and BGN.C may be user defined on the MATERIAL statement and have the defaults shown in Table 3-3.

Note: The default values used in Table 3-3 have been modified from those used in the original Slotboom paper to values suggested by Klaassen [114]. To return to the original Slotboom coefficients (BGN.E=9e-3, BGN.N=1e17 and BGN.C=0.5) the parameter BGN.SLOTBOOM must be set on the MODELS statement in place of BGN.

The variation in bandgap is introduced to the other physical models by subtracting the result of Eq. 3-38 from the bandgap, E_g. In addition an adjustment is also made to the electric field terms in the transport models as described earlier. The adjustment takes the form:

$$\vec{E}_n = -\nabla\left(\psi + \frac{kT_L}{q}\ln n_{ie}\right) \quad 3\text{-}39$$

$$\vec{E}_p = -\nabla\left(\psi - \frac{kT_L}{q}\ln n_{ie}\right) \quad 3\text{-}40$$

The variation in the energy bandgap is also applied partially to the electron affinity, χ. The effective electron affinity, χ_{eff} given as follows:

$$\chi_{eff} = \chi + \Delta E_g \times A_s \quad 3\text{-}41$$

where A_s is a user specifiable asymmetry factor. The user can specify the value of the asymmetry factor using the ASYMMETRY parameter of the MATERIAL statement.

Note: In addition to this in-built model for bandgap narrowing ATLAS allows the use of it's C-interpreter module. A user is allowed to write an external file within which an equation for bandgap narrowing may be specified through a C function. The filename is specified with the parameter F.BGN on the MODEL statement. Further details for the C-interpreter may be found in Appendix A.

Space Charge from Incomplete Ionization, Traps and Defects

Overview

Poisson's equation (Equation 3-1) including the carrier concentrations, the ionized donor and acceptor impurity concentrations N_D^+ and N_A^-, charge due to traps and defects, Q_T, has the form:

$$div(\varepsilon\nabla\psi) = q(n - p - N_D^+ + N_A^-) + Q_T \quad 3\text{-}42$$

In ATLAS the default is to assume full impurity ionization (i.e. $N_D^+ = N_{D,total}$ and $N_A^- = N_{A,total}$), and to set Q_T equal to zero.

ATLAS also provides the options of accounting for incomplete ionization of impurities and accounting for additional charge associated with traps and defects.

Incomplete Ionization of Impurities

ATLAS can account for impurity freeze-out [18] by using Fermi-Dirac statistics with appropriate degeneracy factors for conduction and valence bands, GCB and GVB. The ionized donor and acceptor impurity concentrations are then given by

$$N_D^+ = \frac{N_D}{1 + \mathbf{GCB}\exp\left(\frac{\varepsilon_{F_n} - \mathbf{EDB}}{kT_L}\right)} \quad 3\text{-}43$$

$$N_A^- = \frac{N_A}{1 + \mathbf{GVB}\exp\left(\frac{\mathbf{EAB} - \varepsilon_{F_p}}{kT_L}\right)} \quad 3\text{-}44$$

where EDB is the donor energy level, EAB is the acceptor energy level and N_D and N_A are net compensated n-type and p-type doping, respectively. Net compensated doping is defined as follows:

If

$$N_{total} \equiv (N_{D,total} - N_{A,total}) > 0 \quad 3\text{-}45$$

then

$$N_D = |N_{total}| \text{ and } N_A = 0 \quad 3\text{-}46$$

Otherwise

$$N_D = 0 \text{ and } N_A = \quad | \text{ Ntotal } | \quad) \quad 3\text{-}47$$

The INCOMPLETE parameter of the MODELS statement is used to select incomplete ionization and the parameters .

Table 3-4. User Specifiable Parameters for Equations 3-43 and 3-44		
Statement	**Parameter**	**Units**
MATERIAL	GCB	
MATERIAL	EDB	eV
MATERIAL	GVB	
MATERIAL	EAB	eV

To properly handle incomplete ionization in silicon for high doping levels, a new incomplete ionization model has been added.

The models that form incomplete ionization of impurities given by Equations 3-43 and 3-44 give good physical results for low to moderately doped semiconductors. For heavily (greater than $3x10^{18}/cm^3$) doped semiconductors, these models fail to predict experimental results of complete ionization. For silicon, an optional model has been introduced that better matches experimental results. This model is set by the IONIZ parameter of the MODELS statement.

In this model, the activation energies of the dopants in Equations 3-43 and 3-44 have been modified for doping dependence as given in the following equations:

$$\textbf{EDB}\ (\text{meV}) = 44.0 - 3.6x10^{-8}\ N_D^{1/3} \qquad (N_D < 10^{18}) \qquad 3\text{-}48$$

$$\textbf{EAB}\ (\text{meV}) = 43.8 - 3.037x10^{-5}\ N_A^{1/3} \qquad (N_D < 10^{18}) \qquad 3\text{-}49$$

At doping concentrations above $3x10^{18}$ cm^{-3}, the model predicts complete ionization. At doping concentrations between 10^{18} cm^{-3} and $3x10^{18}$ cm^{-3}, the model predicts a linearly interpolated value between the above expressions and complete ionization.

Low Temperature Simulations

In conjunction with Fermi-Dirac statistics and impurity freeze-out, ATLAS simulates device behavior under low operating temperatures. In general, simulations can be peformed at temperatures as low as 50K without loss of quadratic convergence. Below this temperature, carrier and ionization statistics develop sharp transitions which cause slower convergence. Since many more iterations will probably be required if temperatures below 50K are specified, the ITLIMIT parameter of the METHOD statement should be increased.

Due to the limited exponent range on some machines, ATLAS can have trouble calculating the quasi-Fermi level of minority carriers. As the temperature decreases, the intrinsic carrier concentration also decreases. In quasi-neutral regions, the minority carrier concentration can easily underflow. Such situations were handled in the past by setting these concentrations to zero. This method does not allow an accurate subsequent calculation of minority carrier quasi-Fermi levels. In order to accurately calculate quasi-Fermi levels, majority carrier concentration and the relation, $np=n_{ie}^2$ is used to obtain minority carrier concentrations in case of an underflow. Despite these efforts, spurious 'glitches' are occasionally observed at low temperatures in the minority quasi-Fermi levels.

Traps and Defects

Overview

Semiconductor materials exhibit crystal flaws which can be caused by dangling bonds at interfaces or by the presence of impurities in the substrate. The presence of these defect centers, or traps, in semiconductor substrates may significantly influence the electrical characteristics of the device. Trap centers, whose associated energy lies in a forbidden gap, exchange charge with the conduction and valence bands through the emission and recombination of electrons. The trap centers change the density of space charge in semiconductor bulk and influence the recombination statistics.

Device physics has established the existence of three different mechanism which add to the space charge term in Poissons' equation, in addition to the ionised donor and acceptor impurities. These are interface trapped charge, interface trap states and bulk trap states. Interface trapped charge is modelled as a sheet of charge at the interface and therefore is controlled by the interface boundary condition. Interface traps and bulk traps will add space charge directly into the right hand side of Poisson's equation. This section describes the definition of bulk trap states and the implementation of these bulk trap states into ATLAS for both steady state and transient conditions.

Semiconductor flaws have two possible states which are called “empty” and “full”. When empty a flaw has a particular cross-section for capturing an electron. A flaw can only either emit or capture an electron. When the charge on the centre has been changed by -q by addition of an electron, a flaw is full, and has a new cross section for hole capture. Two basic types of trap have been found to exist; donor-like (electron traps) and acceptor-like traps (hole traps). The charge contained within each type of trap will depend upon whether or not an electron or hole fills the trap.

A donor-like trap, similiar to ionised donor impurites N_D^+, are positively charged and therefore can only capture an electron. This means that donor-like traps are **positive** when **empty** of an electron but are **neutral** when **filled**. An acceptor-like trap, similiar to ionised acceptor impurities N_A^-, are negatively charged so therefore they may only emit an electron. Therefore acceptor-like traps are **negative** when **filled** but are **neutral** when **empty**. In a semiconductor there is a position of the Fermi level corresponding to neutrality. At this condition, those traps above the Fermi level are defined as acceptor-like and those below the Fermi level are donor-like.

Figure 3-1 shows the terminology used within ATLAS to define the type of trap. The position of the trap is defined relative to the conduction or valence bands using E.LEVEL so for instance, an acceptor trap at 0.4eV would be 0.4eV below the conduction band.

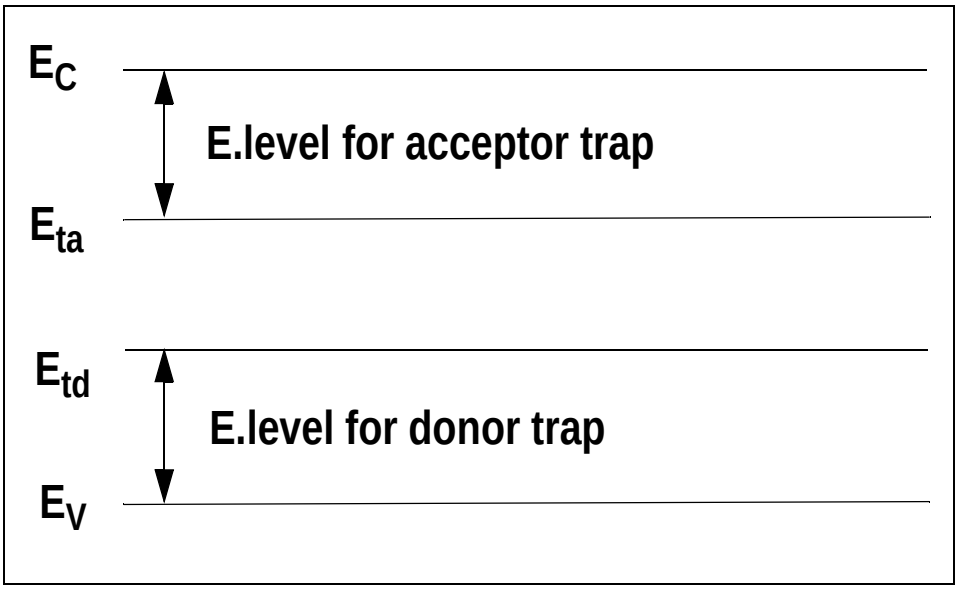

Figure 3-1: Definition of the trap energy level for acceptor and donor traps, in reference to the conduction and valence band edges.

Calculation of Trapped Charge in Poisson's Equation

The total charge caused by the presence of traps is added into the right hand side of Poisson's equation. The total charge value is defined by:

$$Q_T = q(p_t - n_t) \qquad 3\text{-}50$$

where n_t and p_t are the densities of trapped charge for donor-like and acceptor-like traps respectively. The trapped charge depends upon the trap density DENSITY and its probability of occupation, $F_{n,p}$. For donor-like and acceptor-like traps, respectively, the trapped charge is calculated by the equations:

$$n_t = \mathbf{DENSITY} \times F_n \qquad 3\text{-}51$$

$$p_t = \mathbf{DENSITY} \times F_p \qquad 3\text{-}52$$

In the case where multiple traps at multiple trap energy levels are defined the total charge becomes:

$$n_t = \sum_{\alpha=1}^{k} n_t^{\alpha} \quad , \quad p_t = \sum_{\beta=1}^{m} p_t^{\beta} \qquad 3\text{-}53$$

where k is the number of acceptor-like traps and m is the number of donor-like traps.

The probability of occupation assumes that the capture cross sections are constant for all energies in a given band and follows the analysis developed by Simmons and Taylor [126]. The probability of occupation is defined by the following equations for donor- and acceptor-like traps respectively:

$$F_n = \frac{v_n \mathbf{SIGN}\ n + e_p}{v_n(\mathbf{SIGN}\ n + \mathbf{SIGP}\ p) + (e_n + e_p)} \quad 3\text{-}54$$

$$F_p = \frac{v_p \mathbf{SIGP}\ p + e_n}{v_p(\mathbf{SIGN}\ n + \mathbf{SIGP}\ p) + (e_n + e_p)} \quad 3\text{-}55$$

where SIGN and SIGP are the carrier capture cross sections for electrons and holes respectively, v_n and v_p are the thermal velocities for electrons and holes and the electron and hole emission rates, en and ep, are defined by:

$$e_n = \mathbf{DEGEN.FAC}\ v_p\ \mathbf{SIGN}\ n_i \exp\frac{\mathbf{E.LEVEL} - E_i}{kT_L} \quad 3\text{-}56$$

$$e_p = \frac{1}{\mathbf{DEGEN.FAC}}\ v_p\ \mathbf{SIGP}\ n_i \exp\frac{E_i - \mathbf{E.LEVEL}}{kT_L} \quad 3\text{-}57$$

where E_i is the intrinsic Fermi level position, E.LEVEL is the energy level in the bandgap of each discrete trap center and DEGEN.FAC is the degeneracy factor of the trap center. The latter term takes into account that spin degeneracy will exist, that is the “empty” and “full” conditions of a flaw will normally have different spin and orbital degeneracy choices.

Table 3-5. User Specifiable Parameters for Equations 3-51 to 3-57

Statement	Parameter	Units
TRAP	E.LEVEL	eV
TRAP	DENSITY	cm^{-3}
TRAP	DEGEN.FAC	
TRAP	SIGN	cm^2
TRAP	SIGP	cm^2

Trap Implementation into Recombination Models

To maintain self-consistency it is necessary to take into account that electrons are being emitted or captured by the donor- and acceptor-like traps. Therefore the concentration of carriers will be affected. This is accounted for by a modification to the recombination rate in the carrier continuity equations. The standard SRH recombination term is modified as follows:

$$R = \sum_{\alpha=1}^{k} R_n^{\alpha} + \sum_{\beta=1}^{m} R_p^{\beta} \quad 3\text{-}58$$

where k is the number of donor-like traps, m is the number of acceptor-like traps and the function *R* is,

$$R_{n,p} = \frac{pn - n_i^2}{\mathbf{TAUN}\left[p + \frac{1}{\mathbf{DEGEN.FAC}} n_i \exp\frac{E_i - \mathbf{E.LEVEL}}{kT_L}\right] + \mathbf{TAUP}\left[n + \mathbf{DEGEN.FAC}\, n_i \exp\frac{\mathbf{E.LEVEL} - E_i}{kT_L}\right]} \quad 3\text{-}59$$

The electron and hole lifetimes TAUN and TAUP are related to the carrier capture cross sections SIGN and SIGP through the equations

$$\mathbf{TAUN} = \frac{1}{\mathbf{SIGN}\, v_n \mathbf{DENSITY}} \quad 3\text{-}60$$

$$\mathbf{TAUP} = \frac{1}{\mathbf{SIGP}\, v_p \mathbf{DENSITY}} \quad 3\text{-}61$$

Table 3-6. User Specifiable Parameters for Equations 3-59- 3-61

Statement	Parameter	Units
TRAP	TAUN	s
TRAP	TAUP	s

The TRAP statement activates the model and is used to:

- Specify the trap type DONOR or ACCEPTOR
- Specify the energy level E.LEVEL parameter
- Specify the density of the trap centers DENSITY
- Specify the degeneracy factor DEGEN.FAC
- Specify either the cross sections, SIGN and SIGP, or the electron and hole lifetimes TAUN and TAUP

Trap-Assisted Tunneling

At high electric fields, tunneling of electrons from the valence band to the conduction band via trap or defect (trap-assisted tunneling) states can have an important effect on the current. Trap assisted tunneling is modeled by including the field-effect enhancement terms [125], Γ_n and Γ_p in the capture cross-sections. These enhancement terms modify the capture cross-sections so that they include the effects of phonon-assisted tunneling of the emission of electrons and holes from a trap. This model is enabled by specifying TRAP.TUNNEL is specified on the MODELS statement.

The electron and hole capture cross-sections (SIGN and SIGP) in the previous equations are modified in the following manner:

$$\sigma_n = \frac{\mathbf{SIGN}}{1 + \Gamma_n} \quad 3\text{-}62$$

$$\sigma_p = \frac{\mathbf{SIGP}}{1 + \Gamma_p} \quad 3\text{-}63$$

σ_n replaces SIGN in equations 3-54, 3-55, 3-56 and 3-60 and σ_p replaces SIGP in equations 3-54, 3-55, 3-57 and 3-61.

For values of the electric field such that $K_{n,p} > \frac{2\Delta E_{n,p}}{3kT}$ where:

$$K_{n,p} = \frac{8\pi\sqrt{2\ \mathbf{MASS.TUNNEL}\ m_0\ \Delta E_{n,p}^3}}{3qh|E|} \quad 3\text{-}64$$

q is the electronic charge, h is Planck's constant, m_0 is the rest mass of an electron and MASS.TUNNEL is the effective mass (the parameter MASS.TUNNEL may be set on the MODELS statement). The term $\Delta E_{n,p}$ is given by:

$$\Delta E_n = \begin{cases} E_C - E_{fn} & E_T \le E_{fn} \\ E_C - E_T & E_T > E_{fn} \end{cases} \quad 3\text{-}65$$

and

$$\Delta E_p = \begin{cases} E_{fp} - E_V & E_T > E_f \\ E_C - E_T & E_T \le E_f \end{cases} \quad 3\text{-}66$$

where E_C is the conduction band edge, E_V is the valence band edge, E_{fn} is the electron quasi-Fermi level, E_{fp} is the hole quasi-Fermi level and E_T is the energy of the trap level. The electron and hole field enhancement factors then become:

$$\Gamma_{n,p} = 2\sqrt{3\pi}\frac{|E|}{E_\Gamma}\exp\left[\left(\frac{E}{E_\Gamma}\right)^2\right] \quad 3\text{-}67$$

where

$$E_\Gamma = \frac{2\pi\sqrt{24\ \mathbf{MASS.TUNNEL}\ m_0(kT_L)^3}}{qh} \quad 3\text{-}68$$

and k is Boltzmann's constant and T_L is the lattice temperature and E is the electric field.

At high values of electric field where $K_{n,p} \le \frac{2\Delta E_{n,p}}{3kT_L}$ the enhancement terms becomes:

$$\Gamma_{n,p} = \frac{\Delta E_{n,p}}{2kT_L}\sqrt{\frac{\pi}{a}}\exp\left[\frac{b^2 - ac}{a}\right]\mathrm{erfc}\left(\frac{b}{\sqrt{a}}\right) \quad 3\text{-}69$$

where:

$$a = \frac{3}{8}K_{n,p}\ ,\ b = \frac{\Delta E_{n,p}}{2kT_L} - \frac{3}{4}K_{n,p}\ \text{and}\ c = K_{n,p} - \frac{\Delta E_{n,p}}{kT_L} \quad 3\text{-}70$$

Table 3-7. User Specifiable Parameters for Trap-Assisted Tunneling Model			
Statement	**Parameter**	**Default**	**Units**
MODELS	TRAP.TUNNEL	FALSE	
MATERIAL	MASS.TUNNEL	0.25	

Transient Traps

In the time domain the acceptor and donor traps do not reach equilibrium instantaneously but require time for electrons to be emitted or captured. This is taken account of inside ATLAS by solving an additional differential rate equation whenever a transient simulation is performed. The probability of occupation becomes linked to the time dependent emission of electrons through the rate equation:

$$\frac{d}{dt}(\mathbf{DENSITY}\ F_{n,p}) = R_n(t) - R_p(t) \qquad 3\text{-}71$$

where $R_n(t)$ and $R_p(t)$ are the recombination rates of electrons and holes respectively. The terms describing the emission and capture processes R_n and R_p take form:

$$R_n = \mathbf{DENSITY}\ v_n\ \mathbf{SIGN}\left[n\ (1 - F_n) - F_n \mathbf{DEGEN.FAC}\ n_i \exp\left(\frac{\mathbf{E.LEVEL} - E_i}{kT_L}\right)\right] \qquad 3\text{-}72$$

$$R_p = \mathbf{DENSITY}\ v_p\ \mathbf{SIGP}\left[p\ F_p - (1 - F_p)\frac{1}{\mathbf{DEGEN.FAC}}\ n_i \exp\left(\frac{E_i - \mathbf{E.LEVEL}}{kT_L}\right)\right] \qquad 3\text{-}73$$

A transient trap simulation using this model is more time consuming than using the static model but gives a much more accurate description of the device physics. It may sometimes be acceptable to perform transient calculations using the static trap distribution and assume that traps reach equilibrium instantaneously. If this is the case a flag FAST on the TRAP statement will neglect the trap rate equation from the simulation.

The Energy Balance Transport Model

Overview

The conventional drift-diffusion model of charge transport neglects nonlocal transport effects such as velocity overshoot, diffusion associated with the carrier temperature and the dependence of impact ionisation rates on carrier energy distributions. These phenomena can have a significant effect on the terminal properties of submicron devices. As a result ATLAS offers two nonlocal models of charge transport; the energy balance and hydrodynamic models.

The energy balance model follows the derivation by Stratton [48] which is derived starting from the Boltzmann Transport Equation. By applying certain assumptions this model decomposes into the hydrodynamic model.

The energy balance transport model adds continuity equations for the carrier temperatures, and treats mobilities and impact ionization coefficients as functions of the carrier temperatures rather than functions of the local electric field.

The Energy Balance Equations

The energy balance model introduces two new independent variables T_n and T_p, the carrier temperature for electrons and holes. The energy balance equations consist of an energy balance equation with the associated equations for current density and and energy flux $S_{n,p}$.

For electrons the energy balance model consists of:

$$div\vec{S}_n = -\vec{J}_n\nabla\psi - W_n - \frac{3k}{2}\frac{\partial}{\partial t}(\lambda_n^* nT_n^*) \quad 3\text{-}74$$

$$\vec{J}_n = qD_n\nabla n - \mu_n n\nabla\psi + qnD_n^T\nabla T_n \quad 3\text{-}75$$

$$\vec{S}_n = -K_n\nabla T_n - \left(\frac{k\delta_n}{q}\right)\vec{J}_n T_n \quad 3\text{-}76$$

and for holes;

$$div\vec{S}_p = -\vec{J}_p\nabla\psi - W_p - \frac{3k}{2}\frac{\partial}{\partial t}(\lambda_p^* pT_p^*) \quad 3\text{-}77$$

$$\vec{J}_p = qD_p\nabla_p - \mu_p p\nabla\psi + qpD_p^T\nabla T_p \quad 3\text{-}78$$

$$\vec{S}_p = -K_p\nabla T_p - \left(\frac{k\delta_p}{q}\right)\vec{J}_p T_p \quad 3\text{-}79$$

where:

$\vec{S}_n$ and $\vec{S}_p$ are the energy flux densities associated with electrons and holes,

W_n, W_p are the energy density loss rates for electrons and holes,

K_n, K_p are the thermal conductivities of electrons and holes,

D_n, D_p are the thermal diffusivities for electrons and holes, and

μ_n , μ_p are the electron and hole mobilities.

The remaining terms involving the electrons are defined by the following equations:

$$D_n = \frac{\mu_n k T_n}{q}\lambda_n$$

$$\lambda_n = \frac{F(1/2)(\eta_n)}{F-1/2(\eta_n)}, \qquad \eta_n = \frac{\varepsilon_{F_n} - \varepsilon c}{kT_n} = F_{1/2}^{-1}\left(\frac{n}{N_c}\right) \qquad 3\text{-}80$$

$$D_n^T = \left(\mu_{2n} - \frac{3}{2}\lambda_n\mu_n\right)\frac{k}{q}$$

$$\mu_{2n} = \mu_n\left(\frac{5}{2} + \xi_n\right)\frac{F_{\xi_n} + 3/2(\eta_n)}{F_{\xi_n} + 1/2(\eta_n)} \qquad 3\text{-}81$$

$$K_n = q\ n\ \mu_n\left(\frac{k}{q}\right)^2\Delta_n T_n$$

$$\Delta_n = \delta_n\left[\left(\xi_n + \frac{7}{2}\right)\frac{F_{\xi_n} + 5/2(\eta_n)}{F_{\xi_n} + 3/2(\eta_n)} - \left(\xi_n + \frac{5}{2}\right)\frac{F_{\xi_n} + 3/2(\eta_n)}{F_{\xi_n} + 1/2(\eta_n)}\right]$$

$$\delta_n = \frac{\mu_{2n}}{\mu_n} \qquad 3\text{-}82$$

Similar expressions for holes are:

$$D_p = \frac{\mu_p k T_p}{q}\lambda_p$$

$$\lambda_n = \frac{F(1/2)(\eta_n)}{F-1/2(\eta_n)}, \qquad \eta_p = \frac{\varepsilon_v - \varepsilon_{f_p}}{kT_p} = F_{1/2}^{-1}\left(\frac{p}{N_V}\right) \qquad 3\text{-}83$$

$$D_p^T = \left(\mu_{2p} - \frac{3}{2}\lambda_p\mu_p\right)\frac{k}{q}$$

$$\mu_{2p} = \mu_p\left(\frac{5}{2} + \xi_p\right)\frac{F_{\xi_p} + 3/2(\eta_p)}{F_{\xi_p} + 1/2(\eta_p)} \qquad 3\text{-}84$$

$$K_p = q\ p\ \mu_p\left(\frac{k}{q}\right)^2\Delta_p T_p$$

$$\Delta_p = \delta_p\left[\left(\xi_p + \frac{7}{2}\right)\frac{F_{\xi_p} + 5/2(\eta_p)}{F_{\xi_p} + 3/2(\eta_p)} - \left(\xi_p + \frac{5}{2}\right)\frac{F_{\xi_p} + 3/2(\eta_p)}{F_{\xi_p} + 1/2(\eta_p)}\right]$$

$$\delta_p = \frac{\mu_{2p}}{\mu_p} \quad \text{3-85}$$

If Boltzmann statistics are used in preference to Fermi statistics, the above equations simplify to:

$$\lambda_n = \lambda_p = 1 \quad \text{3-86}$$

$$\Delta_n = \delta_n = \left(\frac{5}{2} + \xi_n\right) \quad \text{3-87}$$

$$\Delta_p = \delta_p = \left(\frac{5}{2} + \xi_p\right) \quad \text{3-88}$$

$$\xi_n = \frac{d(\ln\mu_n)}{d(\ln\mu_n)} = \frac{T_n}{\mu_n}\frac{\partial\mu_n}{\partial T_n} \quad \text{3-89}$$

$$\xi_p = \frac{d(\ln\mu_p)}{d(\ln\mu_p)} = \frac{T_p}{\mu_p}\frac{\partial\mu_p}{\partial T_p} \quad \text{3-90}$$

The parameters ξ_n and ξ_p are dependent on the carrier temperatures. Different assumptions concerning ξ_n and ξp correspond to different non-local models. In the high-field saturated-velocity limit, that corresponds to velocity saturation, the carrier mobilities are inversely proportional to the carrier temperatures.

$$\xi_n = \xi_p = -1 \quad \text{3-91}$$

and this corresponds to the Energy Balance model.

If instead the choice ξ_n=ξ_p=0 was chosen this would correspond to the simplified Hydrodynamic model. The parameters ξ_n and ξ_p may be specified using the parameters KSN and KSP on the MODELS statement.

Boundary conditions for n, p and ψ are the same as for the drift diffusion model. Energy balance equations are solved only in the semiconductor region. Electron and hole temperatures are set equal to the lattice temperature on the contacts. On the other part of the boundary, the normal components of the energy fluxes vanish.

Hot carrier transport equations are activated by the MODELS statement parameters HCTE.EL (electron temperature), HCTE.HO (hole temperature), and HCTE (both carrier temperatures).

Density of States

The calculation of the effective density of states is modified for the energy balance model. The electron and hole temperatures replace the lattice temperature in Equations 3-29 and 3-30, i.e.

$$N_c = \left(\frac{2\pi m_e^* kT_n}{h^2}\right)^{\frac{3}{2}} = \left(\frac{T_n}{300}\right)^{\frac{3}{2}} N_c(300) \quad 3\text{-}92$$

$$N_v = \left(\frac{2\pi m_n^* kT_p}{h^2}\right)^{\frac{3}{2}} = \left(\frac{T_p}{300}\right)^{\frac{3}{2}} N_v(300) \quad 3\text{-}93$$

Energy Density Loss Rates

The energy density loss rates define physical mechanisms by which carriers exchange energy with the surrounding lattice environment. These mechanisms include carrier heating with increasing lattice temperature as well as energy exchange through recombination processes (SRH and Auger) and generation processes (impact ionization). If the net generation-recombination rate is written in the form:

$$U = R_{srh} - R_n^A + R_p^A + G_n - G_p \quad 3\text{-}94$$

where R_{srh} is the SRH recombination rate, R_n^A are R_p^A Auger recombination rates related to electrons and holes, G_n and G_p are impact ionization rates, then the energy density loss rates in equations 3-74 and 3-77 can be written in the following form:

$$W_n = \frac{3}{2} n \frac{k(T_n - T_L)}{\mathbf{TAUREL.EL}} \lambda_n^* + \frac{3}{2} kT_n \lambda_n^* R_{SRH} + E_g(G_n - R_n^A) \quad 3\text{-}95$$

$$W_p = \frac{3}{2} p \frac{k(T_p - T_L)}{\mathbf{TAUREL.HO}} \lambda_p^* + \frac{3}{2} k\ T_p\ \lambda_p^* R_{SRH} + E_g(G_p - R_p^A) \quad 3\text{-}96$$

where $\lambda_{n,p}^*$ are defined by expressions 3-80 and 3-83 (and are equal to 1 for Boltzmann statistics), TAUREL.EL and TAUREL.HO are the electron and hole energy relaxation times, E_g is the bandgap energy of the semiconductor. The relaxation parameters are user definable on the MODELS statements and have the default parameters shown in Table 3-8.

The relaxation times are extremely important as they determine the time constant for the rate of energy exchange and therefore precise values are required if the model is to be accurate. However, this is not a measurable parameter and Monte Carlo analysis is the only method through which values may be extracted for the relaxation time.

It is also important to take into consideration that different materials will have different values for the energy relaxation time but within ATLAS the relaxation time will always default to the value for silcon.

Table 3-8. User Specifiable Parameters for Equations 3-95 and 3-96			
Statement	**Parameter**	**Default**	**Units**
MODELS	TAUREL.EL	0.4e-12	s
MODELS	TAUREL.HO	0.4e-12	s

Temperature Dependence of Relaxation Times

ATLAS does not provide an explicit default model for the temperature dependence of energy relaxation times. However two methods exist to make the relaxation time a function of carrier energy.

First, the user can use an in-built model where the parameters τ_{e1}, τ_{e2}, τ_{e3}, W_{e1}, W_{e2}, and W_{e3} in the MATERIAL statement and activate the electron temperature (electron energy) dependent energy relaxation time using E.TAUR.VAR in the MODELS statement. Electron energy relaxation time will then be:

$$\tau_e = \begin{cases} \textbf{TRE.T1}, W < W_{e1} \\ \textbf{TRE.T2}, W = W_{e2} \\ \textbf{TRE.T3}, W > W_{e3} \end{cases} \quad 3\text{-}97$$

where:

$$W = \frac{3}{2} k T_n \quad 3\text{-}98$$

For $W_{e1}<W<W_{e2}$ the energy relaxation time varies quadratically between TRE.T1 and TRE.T2. For We2<W<We3 energy relaxation time varies quadratically between TRE.T2 and TRE.T3. The corresponding parameter for hole energy relaxation time in the MODELS statement is H.TAUR.VAR; other parameters are listed in Table 3-9. The default values are:

$$\textbf{TRE.T1 = TRE.T2 = TRE.T3 = TAUREL.EL} \quad 3\text{-}99$$

$$\textbf{TRH.T1 = TRH.T2 = TRH.T3 = TAUREL.HO} \quad 3\text{-}100$$

Table 3-9. User Specifiable Parameters for Variable Energy Relaxation Time.		
Statement	**Parameter**	**Units**
MATERIAL	TRE.T1	s
MATERIAL	TRE.T2	s
MATERIAL	TRE.T3	s

Table 3-9. User Specifiable Parameters for Variable Energy Relaxation Time.		
Statement	**Parameter**	**Units**
MATERIAL	TRE.W1	eV
MATERIAL	TRE.W2	eV
MATERIAL	TRE.W3	eV
MATERIAL	TRH.T1	s
MATERIAL	TRH.T2	s
MATERIAL	TRH.T3	s
MATERIAL	TRH.W1	eV
MATERIAL	TRH.W2	eV
MATERIAL	TRH.W3	eV

Alternatively, it is possible to use the C-interpreter to apply a user defined model for energy relaxation time as a function of carrier energy. On the MODELS statement the parameter F.TAURN and F.TAURP should be assigned the names of external files that contain the user defined C-interpreter function for the energy relaxation time. The flag E.TAUR.VAR and H.TAUR.VAR should also be specified in the MODELS statement when using C-interpreter functions for the energy relaxation times.

Energy Dependent Mobilities

The energy balance transport model requires the carrier mobility to be related to the carrier energy. This has been achieved through the homogeneous steady state energy balance relationship that pertains in the saturated velocity limit. This allows an effective electric field to be calculated which is the uniform electric field value which causes the carriers in an homogeneous sample to attain the same temperature as at the node point in the device. The effective electric fields $E_{eff,n}$ and $E_{eff,p}$ are calculated by solving the equations:

$$q\mu_n(E_{eff,n})E_{eff,n}^2 = \frac{3}{2}\frac{k(T_n - T_L)}{\mathbf{TAUREL.EL}} \quad 3\text{-}101$$

$$q\mu_p(E_{eff,p})E_{eff,p}^2 = \frac{3}{2}\frac{k(T_p - T_L)}{\mathbf{TAUREL.HO}} \quad 3\text{-}102$$

for $E_{eff,n}$ and $E_{eff,p}$. These equations are derived friom the energy balance equations by stripping out all spatially varying terms. The effective electric fields are then introducecd into the relevent field dependent mobility model. A full description of the available models is given in the mobility section of this chapter.

Boundary Physics

Overview

ATLAS supports several boundary conditions: ohmic contacts, Schottky contacts, insulated contacts, and Neumann (reflective) boundaries. Voltage boundary conditions are normally specified at contacts, but current boundary conditions can also be specified. Additional boundary conditions have been implemented to address the needs of specific applications. Lumped elements can be connected between applied biases and semiconductor device contacts. A true distributed contact resistance is included to account for the finite resistivity of semiconductor contacts.

Ohmic Contacts

Ohmic contacts are implemented as simple Dirichlet boundary conditions, where surface potential, electron concentration, and hole concentrations (ψ_s, n_s, p_s) are fixed. Minority and majority carrier quasi-Fermi potentials are equal to the applied bias of the electrode, i.e. $\phi_n=\phi_p=V_{applied}$. The potential ψ_s is fixed at a value that is consistent with space charge neutrality, i.e.

$$n_s + N_A^- = p_s + N_D^+ \quad \text{3-103}$$

Equation 3-103 can be solved for ψ_s, n_s and p_s, since ϕ_n and ϕ_p are known. If Boltzmann statistics are used, substitution of Equations 3-34 and 3-35 into Equation 3-103 yields:

$$n_s = \frac{1}{2}\left[(N_D^+ - N_A^-) + \sqrt{\left(N_D^+ - N_A^-\right)^2 + 4n_{ie}^2}\right] \quad \text{3-104}$$

$$p_s = \frac{n_{ie}^2}{n_s} \quad \text{3-105}$$

$$\psi_s = \phi_n + \frac{kT_L}{q}\ln\frac{n_s}{n_{ie}} = \phi_p - \frac{kT_L}{q}\ln\frac{p}{n_{ie}} \quad \text{3-106}$$

Note: Contacts are assumed to be ohmic if no workfunction is specified in ATLAS, irrespective of the material of the contact.

Schottky Contacts

Schottky contacts are defined by the work function of the electrode metal and an optional surface recombination velocity. The surface potential at a Schottky contact is defined by:

$$\psi_s = \mathbf{AFFINITY} + \frac{E_g}{2q} + \frac{kT_L}{2q}\ln\frac{N_C}{N_V} - \mathbf{WORKFUN} + V_{applied} \quad 3\text{-}107$$

where AFFINITY is the electron affinity of the semiconductor material, E_g is the bandgap, N_C is the conduction band density of states, N_V is the valence band density of states, and T_L is the ambient temperature.

Table 3-10. User Specifiable Parameters for Equation 3-107		
Statement	**Parameter**	**Units**
CONTACT	WORKFUN	eV
MATERIAL	AFFINITY	eV

A finite surface recombination velocity can be imposed at a contact by specifying the parameter SURF.REC on the CONTACT statement. In this case the quasi-Fermi levels, ϕ_n and ϕ_p, are no longer equal to $V_{applied}$. Instead, these parameters are defined by a current boundary condition at the surface [31]:

$$J_{sn} = q\mathbf{VSURFN}(n_s - n_{eq}) \quad 3\text{-}108$$

$$J_{sp} = q\mathbf{VSURFP}(p_s - p_{eq}) \quad 3\text{-}109$$

Table 3-11. User Specifiable Parameters for Equations 3-108 to 3-109.		
Statement	**Parameter**	**Units**
CONTACT	VSURFN	cm/s
CONTACT	VSURFP	cm/s

where J_{sn} and J_{sp} are the electron and hole currents at the contact, n_s is the surface electron concentration and p_s is the surface hole concentrations. The terms n_{eq} and p_{eq} are the equilibrium electron and hole concentrations assuming infinite surface recombination velocity ($\phi_n = \phi_p = V_{applied}$). If v_{sn} and v_{sp} are not specified, their values are calculated from Equations 3-110 and 3-111.

$$v_{sn} = \frac{\mathbf{ARICHN}\ T_L^2}{q\ N_C} \quad 3\text{-}110$$

$$v_{sp} = \frac{\mathbf{ARICHP}\ T_L^2}{q\ N_V} \quad 3\text{-}111$$

Table 3-12. User Specifiable Parameters for Equations 3-110 to 3-111.			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	ARICHN	110	$A/cm^2/K^2$
MATERIAL	ARICHP	30	$A/cm^2/K^2$

where ARICHN and ARICHP are the effective Richardson constants for electrons and holes, account of quantum mechanical reflections and tunneling, N_C and N_V are the conduction and valence band density of states. The parameters ARICHN and ARICHP are user definable as shown in Table 3-12 and $N_C$and N_V are functions of the lattice temperature, T_L, according to equations 3-29 and 3-30.

The Schottky model also accounts for field-dependent barrier-lowering mechanisms. These mechanisms are caused by image forces and possible static dipole layers at the metal-semiconductor interface [13]. If the barrier height is defined as:

$$\phi_{bn} = \mathbf{WORKFUN} - \mathbf{AFFINITY} \quad 3\text{-}112$$

$$\phi_{bp} = \mathbf{AFFINITY} + \frac{E_g}{q} - \mathbf{WORKFUN} \quad 3\text{-}113$$

the amount by which these barriers are lowered becomes:

$$\Delta\phi_b = \left[\frac{q}{4\pi\varepsilon_s}\right]^{\frac{1}{2}} E^{1/2} + \mathbf{ALPHA}\ E \quad 3\text{-}114$$

Table 3-13. User Specifiable Parameters for Equation 3-114.		
Statement	**Parameter**	**Units**
CONTACT	ALPHA	cm

where E is the magnitude of the electric field at the interface and ALPHA is the linear, dipole barrier lowering coefficient.

This barrier lowering coefficient is specified by the ALPHA parameter in the CONTACT statement. Typical values of α may be seen in. Note that the term with the square root dependence on electric field corresponds to the image force, while the linear term corresponds to the dipole effect [13].

Barrier lowering had been previously implemented in an earlier version of S-PISCES [32]. It has been reincorporated into ATLAS in a slightly different manner. In ATLAS, the Poisson equations are normally solved using the boundary condition $\Psi s = \Psi s_0$ (see Equation 3-107).

If electric field is consistent with solved potentials, effective surface potentials are computed as:

$$\psi_{seff} = \psi_{s0} \pm \Delta\phi_b$$

where + is used for electrons, and – is used for holes.

Continuity equations are then solved using Equations 3-108 and 3-109 as boundary conditions. The variables n_s and n_{eq} (see Equation 3-108) are replaced by n_{seff} and n_{eqeff} which are computed using Ψ_{seff}.

The physical interpretation of this approach is that the Poisson equation is solved consistent with the charge, but that electrons and holes use combined Poisson and image-force potential. Although the full barrier lowering term has been applied directly at the surface, this peak is slightly different within the semiconductor.

Surface recombination is implemented on a triangle-by-triangle basis. That is, using the surface recombination velocity and geometrical data, a recombination component is calculated for each triangle so that an element of interest is connected. Using the electric field for each triangle, an adjusted recombination term can be computed if barrier lowering is incorporated. This is in contrast to where a single field value for the electrode node is used to compute total recombination value. Also unlike [32], barrier lowering can be used with any of the basic numerical solution procedures, i.e. Gummel or Newton.

Floating Contacts

A contact that is not connected to a either a current or voltage source and is totally insulated by dielectric is called a floating contact. ATLAS accounts for floating contacts such as floating gates in EPROM devices by using a distributed charge boundary condition which is applied to all nodes of the floating electrode:

$$\int_s D \, dS = Q_{FG} \qquad 3\text{-}115$$

where:

- D is the electric displacement vector.
- S represents the external surface of the floating gate.
- Q_{FG} is the injected charge.

ATLAS performs an integration over the entire surface of the electrode and forces the potential on the floating nodes to produce the correct total charge on the electrode. The total charge when performing a simulation is by default zero but can be defined using the SOLVE statement. The total charge may change if a charge injection model has been turned on. These models are defined in a later section underneath Gate Current Models. To define a contact as a floating contact, use:

```
CONTACT NAME=fgate FLOATING
```

Note: When specifying a floating contact it is necessary to use the Newton scheme as the numerical technique.

Current Boundary Conditions

In some devices, the terminal current is a multi-valued function of the applied voltage. This means that for some voltage boundary conditions, the solution that is obtained depends on the initial guess. An example of this is the CMOS latch-up trigger point. At the trigger point the I-V curve changes from being flat to vertical and may exhibit a negative slope. The solution will then have three different solutions of current for one applied bias. The particular solution which the model finishes in will depend upon the initial conditions.

This trigger point is difficult to determine using a simple voltage boundary condition. In addition, it is almost impossible to compute any solutions in the negative resistance regime when using voltage boundary conditions. Some of these problems can be overcome using current boundary conditions.

Calculation of current boundary conditions is activated by the CURRENT parameter in the CONTACT statement.

The voltage boundary condition should be used in regions where dI/dV is small. The current boundary condition may be preferable for operating regimes where dI/dV is large. It is not uncommon for the negative resistance regime of a device to have a slope dI/dV very close to 0. Such behavior should be considered when using a current source to trace out an entire I-V curve. The CURVETRACE option in ATLAS would be preferable in these cases.

Note: When a current boundary condition has been specified it is necessary to choose the NEWTON numerical scheme on the METHOD statement. It is possible to perform ac small signal analysis with current boundary conditions.

Insulating Contacts

Insulating contacts are contacts that are completely surrounded by insulator. They may be connected to a voltage source (tied contact) or they may be floating.

Insulating contacts that are connected to a voltage source generally have a work function that dictates a value for ψ_s similar to that shown in Equation 3-107. Electron and hole concentrations within the insulator and at the insulating contact are forced to be zero, i.e. $n_s=p_s=0$.

Neumann Boundaries

Along the outer (non-contact) edges of devices, homogeneous (reflecting) Neumann boundary conditions are imposed so that current only flows out of the device through the contacts. In the absence of surface charge along such edges, the normal electric field component becomes zero. Current is not permitted to flow from the semiconductor into an insulating region except via oxide tunneling models. At the interface between two different materials, the difference between the normal components of the respective electric displacements must be equal to any surface charge according to:

$$\hat{n} \cdot \varepsilon_1 \nabla\psi_1 - \hat{n} \cdot \varepsilon_2 \nabla\psi_2 = \rho_s \qquad 3\text{-}116$$

where n is the unit normal vector, ε_1 and ε_2 are the permittivities of the materials on either side of the interface and ρ_s is the sheet charge at the interface.

Lumped R, L and C Elements

ATLAS supports some simple configurations of lumped elements. These are indicated in Figure 3-2.

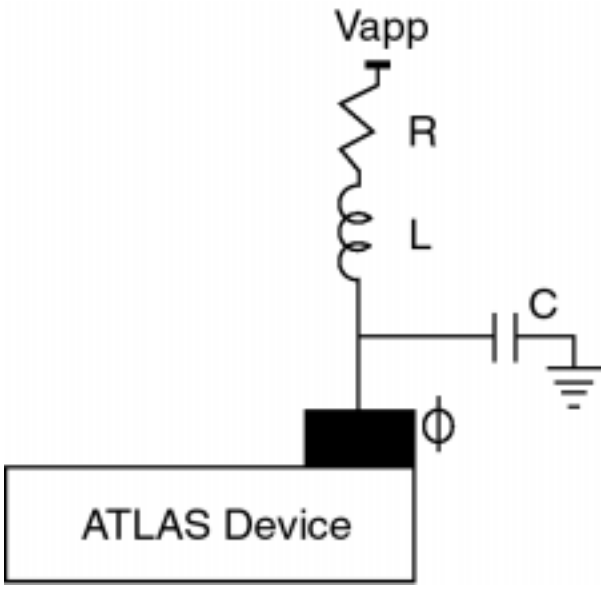

Figure 3-2: The lumped elements supported by ATLAS

Lumped elements can be extremely useful when simulating CMOS or MOSFET structures. For example, the p-tub contact in the CMOS cross-section might be tens or hundreds of microns away from the active area of an embedded vertical npn bipolar transistor, which may only be 10-20 µm on a side. If the whole structure were simulated, a tremendous number of grid points (probably more than half) are essentially wasted in accounting for a purely resistive region of the device.

In the case of a MOSFET substrate, you would not want to include grid points all the way to the back side of a wafer. In either of these cases, a simple lumped resistance can be substituted.

Table 3-14. User Specifiable Parameters for Figure 3-1.

Symbol	Statement	Parameter	Units
C	CONTACT	CAPACITANCE	F/µm
R	CONTACT	RESISTANCE	Ωµm
L	CONTACT	INDUCTANCE	Hµm

Resistance is specified in Ω·m, capacitance is specified in F/µm, and inductance is specified in H·µm.

The following combinations are possible: resistance only; resistance and capacitance; inductance and resistance; inductance, capacitance and resistance. For the case of inductance or capacitance only, ATLAS adds a small resistance as well. For more complicated circuit configurations it is necessary to use the MIXEDMODE module of ATLAS.

Note: Capacitance increases with device width (into the z-plane) while resistance decreases. Except for the case of extremely large resistances where the arrangement becomes similar to a pure current

source, no convergence degradation has been observed for a lumped element boundary in comparison to a simple ohmic contact. Transient simulation therefore becomes easier and is more well-defined.

It is a good idea to use the simulator to calculate any resistance (or capacitance) components that might be included as lumped elements. When peforming CMOS simulations, you could simulate just the p-tub with ohmic contacts at either end. From the plot of terminal current (in A/μm) versus voltage, resistance can be directly extracted from the slope. Be very careful to consider any three-dimensional effects (e.g., current spreading) before using a resistance value in further simulations.

When looking at the results of simulation with lumped elements it is important to distinguish between the applied voltage (V_{app}) and the internal bias (ϕ) in the log file produced by ATLAS.

Note: AC small signal analysis can not be performed when any lumped elements have been specified. AC small signal analysis can be performed with the resistance specified on the LOG statement. Also, when lumped elements have been defined the **NEWTON** numerical scheme should be chosen on the METHOD statement.

Distributed Contact Resistance

Because contact materials have finite resistivities, the electrostatic potential is not always uniform along the metal-semiconductor surface. To account for this effect, a distributed contact resistance can be associated with any electrode. This is in contrast to the lumped contact resistance described in the previous section.

ATLAS implements this distributed contact resistance by assigning a resistance value to each surface element associated with the contact. If each surface element is a different length, a different value for the contact resistance will be assigned. ATLAS calculates a resistance value R_i for each surface element from the value of CON.RESIST as specified on the CONTACT statement. The units of CON.RESIST are Ωcm^2 and R_i calculated as:

$$R_i = \frac{\mathbf{CON.RESIST}}{d_i \ \mathbf{WIDTH}} \qquad 3\text{-}117$$

where R_i is the resistance at node i, P_c is specified by the CON.RESIST parameter, d_i is the length of the contact surface segment associated with node i and WIDTH is the width of the device. The effect of the resistance R_i is to add an extra equation to be satisfied to node i. This equation is given by:

$$\frac{1}{R_i}\left[V_{applied} - \left(\psi_i \pm \frac{kT_L}{q}\ln(N/n_{ie})\right)\right] - (I_n + I_p + I_{disp}) = 0 \qquad 3\text{-}118$$

where $V_{applied}$ is the external applied voltage, Ψ_i is the surface potential, N is the net doping, n_i is the intrinsic electron concentration, and I_n, I_p, I_{disp} are the electron, hole and displacement currents at node i. This equation simply balances the current in and out of the resistor added to each node i.

As with the case for lumped elements, ATLAS can print out a value of contact resistance for each contact in the run time output. Since the actual value depends on the length of each surface segment for distributed contacts, ATLAS prints out the value of CON.RESIST/WIDTH which is the same for all contact surface segments. This runtime output is enabled by adding the PRINT option on the MODELS statement.

Table 3-15. User Specifiable Parameters for Equation 3-117.		
Statement	**Parameter**	**Units**
CONTACT	CON.RESIST	Ω-cm^2
MESH	WIDTH	μm

Note: AC small signal analysis can not be performed when any distributed contact resistance has been specified. Also, only the **NEWTON** numerical scheme should be chosen on the METHOD statement.

Energy Balance Boundary Conditions

When the energy balance model is applied special boundary conditions are applied for carrier temperatures. At the contacts Dirichlet boundary conditions are used for carrier temperatures:

$$T_n = T_p = T_L$$

Elsewhere on the boundary the normal components of the energy fluxes vanish. The boundary conditions for (ψ, n, p) are the same as for the drift-diffusion model.

Physical Models

Mobility Modeling

Overview

Electrons and holes are accelerated by electric fields, but lose momentum as a result of various scattering processes. These scattering mechanisms include lattice vibrations (phonons), impurity ions, other carriers, surfaces, and other material imperfections. Since the effects of all of these microscopic phenomena are lumped into the macroscopic mobilities introduced by the transport equations these mobilities are therefore functions of the local electric field, lattice temperature, doping concentration, and so on.

Mobility modeling is normally divided into: (i) low field behaviour, (ii) high field behavior, (iii) bulk semiconductor regions and (iv) inversion layers.

The low electric field behaviour has carriers almost in equilibrium with the lattice and the mobility has a characteristic low-field value that is commonly denoted by the symbol $\mu_{n0,p0}$. The value of this mobility is dependent upon phonon and impurity scattering. Both of which act to decrease the low field mobility.

The high electric field behaviour shows that the carrier mobility declines with electric field because the carriers that gain energy can take part in a wider range of scattering processes. The mean drift velocity no longer increases linearly with increasing electric field, but rises more slowly. Eventually, the velocity does not increase any more with increasing field, but saturates at a constant velocity. This constant velocity is commonly denoted by the symbol v_{sat}. Impurity scattering is relatively insignificant for energetic carriers, and so v_{sat} is primarily a function of the lattice temperature.

Modeling mobility in bulk material involves: (i) characterizing μ_{n0} and μ_{p0} as a function of doping and lattice temperature, (ii) characterizing v_{sat} as a function of lattice temperature, and (iii) describing the transition between the low field mobility and saturated velocity regions.

Modeling carrier mobilities in inversion layers introduces additional complications. Carriers in inversion layers are subject to surface scattering, extreme carrier-carrier scattering, and quantum mechanical size quantization effects. These effects must be accounted for in order to perform accurate simulation of MOS devices. The transverse electric field is often used as a parameter that indicates the strength of inversion layer phenomena.

It is possible to define multiple non-conflicting mobility models simultaneously. It is also necessary to know which models are over-riding when conflicting models are defined.

Low Field Mobility Models

The low field carrier mobility may be defined in five different ways;

- using the parameters `MUN` and `MUP` to set constant constant values for electron and hole mobilities
- using a look-up table model `CONMOB` to relate the low field mobility at 300K to the impurity concentration
- choice of analytic low field mobility models `ANALYTIC` and `ARORA` to relate the low field carrier mobility to impurity concentration and temperature
- choice of a carrier-carrier scattering model `CCSMOB` that relates the low field mobility to the carrier concentrations and temperature
- a unified low field mobility model `KLAASSEN` that relates the low field mobility to donor, acceptor, lattice, carrier-carrier scattering and temperature

Constant Mobility

In ATLAS the choice of mobility model is specified on the MODELS statement. The parameters associated with mobility models are specified on a separate MOBILITY statement. One or more mobility models should always be specified explicitly. The default is to use constant low field mobilities within each region of a device. This default model is independent of doping concentration, carrier densities and electric field. It does account for lattice scattering due to temperature according to:

$$\mu_{n0} = \mathbf{MUN}\left(\frac{T_L}{300}\right)^{-\mathbf{TMUN}} \quad 3\text{-}119$$

$$\mu_{p0} = \mathbf{MUP}\left(\frac{T_L}{300}\right)^{-\mathbf{TMUP}} \quad 3\text{-}120$$

where T is the lattice temperature and the low field mobility parameters, MUN, MUP, TMUN and TMUP may be specified on the MOBILITY statement with the defaults as shown in Table 3-16.

Table 3-16. User Specifiable Parameters for the Constant Low Field Mobility Model

Statement	Parameter	Default	Units
MOBILITY	MUN	1000	$cm^2/(V \cdot s)$
MOBILITY	MUP	500	$cm^2/(V \cdot s)$
MOBILITY	TMUN	1.5	
MOBILITY	TMUP	1.5	

Note: This particular default leads to unrealistically high carrier velocities at intermediate and high electric fields.

Concentration Dependent Low Field Mobility Tables

ATLAS provides empirical data for the doping dependent low-field mobilities of electrons and holes in silicon at T_L=300K only. This data is used if the CONMOB parameter is specified in the MODELS statement. The data that is used is shown in Table 3-17.

Table 3-17. Mobility of Electrtons and Holes in Silicon at T=300K

Concentration (cm^{-3})	Mobility (cm^2/v-s)	
	Electrons	Holes
$1.0*10^{14}$	1350.0	495.0
$2.0*10^{14}$	1345.0	495.0
$4.0*10^{14}$	1335.0	495.0

Table 3-17. Mobility of Electrtons and Holes in Silicon at T=300K (Continued)

Concentration (cm^{-3})	Mobility (cm^2/v-s)	
	Electrons	Holes
$6.0*10^{14}$	1320.0	495.0
$8.0*10^{14}$	1310.0	495.0
$1.0*10^{15}$	1300.0	491.1
$2.0*10^{15}$	1248.0	487.3
$4.0*10^{15}$	1200.0	480.1
$6.0*10^{15}$	1156.0	473.3
$8.0*10^{15}$	1115.0	466.9
$1.0*10^{16}$	1076.0	460.9
$2.0*10^{16}$	960.0	434.8
$4.0*10^{16}$	845.0	396.5
$6.0*10^{16}$	760.0	369.2
$8.0*10^{16}$	720.0	348.3
$1.0*10^{17}$	675.0	331.5
$2.0*10^{17}$	524.0	279.0
$4.0*10^{17}$	385.0	229.8
$6.0*10^{17}$	321.0	2103.8
$8.0*10^{17}$	279.0	186.9
$1.0*10^{18}$	252.0	178.0
$2.0*10^{18}$	182.5	130.0
$4.0*10^{18}$	140.6	90.0
$6.0*10^{18}$	113.6	74.5
$8.0*10^{18}$	99.5	66.6
$1.0*10^{19}$	90.5	61.0
$2.0*10^{19}$	86.9	55.0
$4.0*10^{19}$	83.4	53.7
$6.0*10^{19}$	78.8	52.9

Table 3-17. Mobility of Electrtons and Holes in Silicon at T=300K (Continued)

Concentration (cm^{-3})	Mobility (cm^2/v-s)	
	Electrons	Holes
$8.0*10^{19}$	71.6	52.4
$1.0*10^{20}$	67.8	52.0
$2.0*10^{20}$	52.0	50.8
$4.0*10^{20}$	35.5	49.6
$6.0*10^{20}$	23.6	48.9
$8.0*10^{20}$	19.0	48.4
$1.0*10^{21}$	17.8	48.0

The Analytic Low Field Mobility Model

The following analytic function based upon the work of Caughey and Thomas [22] can be used to specify doping- and temperature-dependent low-field mobilities.

$$\mu_{n0} = \mathbf{MU1N.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathbf{ALPHAN.CAUG}} + \frac{\mathbf{MU2N.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathbf{BETAN.CAUG}} - \mathbf{MU1N.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathbf{ALPHAN.CAUG}}}{1 + \left(\frac{T_L}{300K}\right)^{\mathbf{GAMMAN.CAUG}} \cdot \left(\frac{N}{\mathbf{NCRITN.CAUG}}\right)^{\mathbf{DELTAN.CAUG}}} \quad \text{3-121}$$

$$\mu_{p0} = \mathbf{MU1P.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathbf{ALPHAP.CAUG}} + \frac{\mathbf{MU2P.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathbf{BETAP.CAUG}} - \mathbf{MU1P.CAUG} \cdot \left(\frac{T_L}{300K}\right)^{\mathbf{ALPHAP.CAUG}}}{1 + \left(\frac{T_L}{300K}\right)^{\mathbf{GAMMAP.CAUG}} \cdot \left(\frac{N}{\mathbf{NCRITP.CAUG}}\right)^{\mathbf{DELTAP.CAUG}}} \quad \text{3-122}$$

where N is the local (total) impurity concentration in cm^{-3} and T_L is the temperature in degrees Kelvin.

This model is activated by specifying both the CONMOB and ANALYTIC parameters in the MODELS statement. The parameters of this model are specified in the MOBILITY statement. The default parameters are for silicon at T_L = 300K.

Table 3-18. User Specifiable Parameters for Equations 3-121 and 3-122

Statement	Parameter	Default	Units
MOBILITY	MU1N.CAUG	55.24	$cm^2/(V \cdot s)$
MOBILITY	MU1P.CAUG	49.7	$cm^2/(V \cdot s)$
MOBILITY	MU2N.CAUG	1429.23	$cm^2/(V \cdot s)$
MOBILITY	MU2P.CAUG	479.37	$cm^2/(V \cdot s)$
MOBILITY	ALPHAN.CAUG	0.0	unitless
MOBILITY	ALPHAP.CAUG	0.0	unitless
MOBILITY	BETAN. CAUG	-2.3	unitless
MOBILITY	BETAP. CAUG	-2.2	unitless
MOBILITY	GAMMAN.CAUG	-3.8	unitless
MOBILITY	GAMMAP.CAUG	13.7	unitless
MOBILITY	DELTAN.CAUG	0.73	unitless
MOBILITY	DELTAP.CAUG	0.70	unitless
MOBILITY	NCRITN.CAUG	$1.072X10^{17}$	cm^{-3}
MOBILITY	NCRITP.CAUG	$1.606X10^{17}$	cm^{-3}

The Arora Model for Low Field Mobility

Another analytic model for the doping and temperature dependence of the low field mobility is available in ATLAS. This model, which is due to Arora et. al. [119] has the following form:

$$\mu_{n0} = \mathbf{MU1N.ARORA}\left(\frac{T_L}{300}\right)^{\mathbf{ALPHAN.ARORA}} + \frac{\mathbf{MU2N.ARORA}\left(\frac{T_L}{300}\right)^{\mathbf{BETAN.ARORA}}}{1 + \frac{N}{\mathbf{NCRITN.ARORA} \cdot \left(\frac{T_L}{300}\right)^{\mathbf{GAMMAN.ARORA}}}} \quad \text{3-123}$$

$$\mu_{p0} = \mathbf{MU1P.ARORA}\left(\frac{T_L}{300}\right)^{\mathbf{ALPHAP.ARORA}} + \frac{\mathbf{MU2P.ARORA}\left(\frac{T_L}{300}\right)^{\mathbf{BETAP.ARORA}}}{1 + \frac{N}{\mathbf{NCRITP.ARORA} \cdot \left(\frac{T_L}{300}\right)^{\mathbf{GAMMAP.ARORA}}}} \quad \text{3-124}$$

This model is used if CONMOB and ARORA are specified on the MODELS statement. The parameters of the model are specified in the MOBILITY statement. The default parameters are for silicon at T_L=300K.

Table 3-19. User Specifiable Parameters for Equations 3-123 and 3-124			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MU1N.ARORA	88.0	$cm^2/(V \cdot s)$
MOBILITY	MU1P.ARORA	54.3	$cm^2/(V \cdot s)$
MOBILITY	MU2N.ARORA	1252.0	$cm^2/(V \cdot s)$
MOBILITY	MU2P.ARORA	407.0	$cm^2/(V \cdot s)$
MOBILITY	ALPHAN.ARORA	-0.57	unitless
MOBILITY	ALPHAP.ARORA	-0.57	unitless
MOBILITY	BETAN.ARORA	-2.33	unitless
MOBILITY	BETAP.ARORA	-2.23	unitless
MOBILITY	GAMMAN.ARORA	2.546	unitless
MOBILITY	GAMMAP.ARORA	2.546	unitless
MOBILITY	NCRITN.ARORA	$1.432X10^{17}$	cm^{-3}
MOBILITY	NCRITP.ARORA	$2.67X10^{17}$	cm^{-3}

The Carrier-Carrier Scattering Model For Low Field Mobility

The Dorkel and Leturq [66] model for low field mobility includes the dependence on temperature, doping, and carrier-carrier scattering. This model is activated by specifying the CCSMOB parameter of the MODELS statement. The parameters of the model are specified in the MOBILITY statement. This model has the form:

$$\mu_{n0,\,p0} = \mu^{L}_{n,\,p}\left(\frac{1.025}{1+\left[1.126\left(\frac{\mu^{L}_{n,\,p}}{\mu^{IC}}\right)\right]^{0.715}}\right)^{-0.025} \qquad 3\text{-}125$$

where:

L is the lattice scattering.
I is the ionized impurity scattering.
C is the carrier-carrier scattering.

$$\mu^{C} = \frac{1.04 \cdot 10^{21}\left(\frac{T_L}{300}\right)^{3/2}}{\sqrt{np}\ln\left[1 + 7.45 \cdot 10^{13}\left(\frac{T_L}{300}\right)^{2}(np)^{-1/3}\right]} \quad \text{3-126}$$

$$\mu_n^{I} = \frac{\frac{\mathbf{AN.CCS}}{\mathbf{CCS.EA}}\left(\frac{T_L}{300}\right)^{3/2}}{N_T} f\left[\frac{\mathbf{BN.CCS}\left(\frac{T_L}{300}\right)^{2}}{n+p}\right] \quad \text{3-127}$$

$$\mu_p^{I} = \frac{\mathbf{AP.CCS}\left(\frac{T_L}{300}\right)^{3/2}}{N_T} f\left[\frac{\mathbf{BP.CCS}\left(\frac{T_L}{300}\right)^{2}}{n+p}\right] \quad \text{3-128}$$

where:

$$f(x) = \left[\ln(1+x) - \frac{x}{1+x}\right]^{-1}$$

The values of the lattice scattering terms $\mu_{n,p}{}^{L}$ are defined by:

$$\mu_n^{L} = \mathbf{MUN}\left(\frac{T_L}{300}\right)^{-2.2} \quad \text{3-129}$$

$$\mu_p^{L} = \mathbf{MUP}\left(\frac{T_L}{300}\right)^{-2.2} \quad \text{3-130}$$

Table 3-20. User Specifiable Parameters for Equations 3-127 and 3-128..

Statement	Parameter	Default	Units
MOBILITY	AN.CCS/CCS.EA	$4.61*10^{17}$	cm^{-3}
MOBILITY	AP.CCS	$1.0*10^{17}$	cm^{-3}
MOBILITY	BN.CCS	$1.52*10^{15}$	cm^{-3}
MOBILITY	BP.CCS	$6.25*10^{14}$	cm^{-3}

Klaassen's Unified Low Field Mobility Model

The model by D. B. M. Klaassen [114, 115], provides a unified description of majority and minority carrier mobilities. In so doing, it includes the effects of lattice scattering, impurity scattering (with screening from charged carriers), carrier-carrier scattering, and impurity clustering effects at high concentration. The model shows excellent agreement between the modeled and empirical data for:

- majority electron mobility as a function of donor concentration over the range of 10^{14} cm^{-3} to 10^{22} cm^{-3}
- minority electron mobility as a function of acceptor concentration over the range of 10^{17} cm^{3} to 10^{20} cm^{-3}
- minority hole mobility as a function of donor concentration from 10^{17} cm^{-3} to 10^{20} cm^{-3}
- temperature dependence over the range of 70 K to 500 K

The Klaassen model accounts for a broader set of effects and has been calibrated over a wider range of conditions than any other of the low field bulk mobility models. As such it is the recommended model for both MOS and bipolar simulation and is the default model for silicon. The model can be enabled or disabled using the KLA parameter on the MODELS statement, or independently for electrons and holes by the KLA.N and KLA.P parameters of the MOBILITY statement.

The total mobility can be described by its components using Mathiesen's rule as:

$$\mu_{n0}^{-1} = \mu_{nL}^{-1} + \mu_{nDAP}^{-1} \quad \text{3-131}$$

$$\mu_{p0}^{-1} = \mu_{pL}^{-1} + \mu_{pDAP}^{-1} \quad \text{3-132}$$

μ_n and μ_p are the total low field electron and hole mobilities, μ_{nL} and μ_{pL} are the electron and hole mobilities due to lattice scattering, μ_{nDAP} and μ_{pDAP} are the electron and hole mobilities due to donor(D), acceptor(A), screening(P) and carrier-carrier scattering.

The lattice scattering components, μ_{nL} and μ_{pL} are given as:

$$\mu_{nL} = \mathbf{MUMAXN.KLA}\left(\frac{300}{T_L}\right)^{\mathbf{THETAN.KLA}} \quad \text{3-133}$$

$$\mu_{pL} = \mathbf{MUMAXP.KLA}\left(\frac{300}{T_L}\right)^{\mathbf{THETAP.KLA}} \quad \text{3-134}$$

where T_L is the temperature in degrees Kelvin, MUMAXN.KLA, MUMAXP.KLA, THETAN.KLA and THETAP.KLA are user definable model parameters which can be specified as shown in Table 3-18

Table 3-21. User Specifiable Parameters for Equations 3-133 and 3-134

Statement	Parameter	Default	Units
MOBILITY	MUMAXN.KLA	1417.0	$cm^2/(V*s)$

Table 3-21. User Specifiable Parameters for Equations 3-133 and 3-134			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MUMAXP.KLA	470.5	$cm^2/(V*s)$
MOBILITY	THETAN.KLA	2.285	
MOBILITY	THETAP.KLA	2.247	

The impurity-carrier scattering components of the total mobility (Eqs. 3-131 and 3-132) are given by:

$$\mu_{nDAP} = \mu_{N,n}\frac{N_{ncs}}{N_{nsc,eff}}\left(\frac{\mathbf{NREF1N.KLA}}{N_{nsc}}\right)^{\mathbf{ALPHA1N.KLA}} + \mu_{nc}\left(\frac{n+p}{N_{nsc,eff}}\right) \qquad 3\text{-}135$$

$$\mu_{pDAP} = \mu_{N,p}\frac{N_{nsc}}{N_{psc,eff}}\left(\frac{\mathbf{NREF1P.KLA}}{N_{psc}}\right)^{\mathbf{ALPHA1P.KLA}} + \mu_{pc}\left(\frac{n+p}{N_{psc,eff}}\right) \qquad 3\text{-}136$$

where $\mu_{n,N}$ and $\mu_{p,N}$ are the impurity scattering components of mobility and are given by Eq.s. 3-137 and 3-138, μ_{nc} and μ_{pc} are the carrier-carrier scattering component of mobility given by Eqs. 3-139 amd 3-140, N_{nsc} and N_{psc} are given by Eqs. 3-141 and 3-142, and $N_{nsc,eff}$ and $N_{psc,eff}$ are given by Eqs 3-143 and 3-144. The parameters NREF1N.KLA, NREF1P.KLA, ALPHA1N.KLA and ALPHA1P.KLA are user definable model parameters which can be specified as in Table 3-22.

Table 3-22. User Specifiable Parameters for Equation2 3-135 and 3-136			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	ALPHA1N.KLA	0.68	
MOBILITY	ALPHA1P.KLA	0.719	
MOBILITY	NREF1N.KLA	9.68e16	cm^3
MOBILITY	NREF1P.KLA	2.23e17	cm^3

The impurity scattering components, $\mu_{N,n}$ and $\mu_{N,p}$, are given by:

$$\mu_{N,n} = \frac{\mathbf{MUMAXN.KLA}^2}{\mathbf{MUMAXN.KLA} - \mathbf{MUMINN.KLA}}\left(\frac{T_L}{300}\right)^{3\mathbf{ALPHA1N.KLA} - 1.5} \qquad 3\text{-}137$$

$$\mu_{N,p} = \frac{\mathbf{MUMAXP.KLA}^2}{\mathbf{MUMAXP.KLA} - \mathbf{MUMINP.KLA}}\left(\frac{T_L}{300}\right)^{3\mathbf{ALPHA1P.KLA} - 1.5} \qquad 3\text{-}138$$

where T_L is the temperature in degrees Kelvin, MUMINN.KLA and MUMINP.KLA are user defined parameters shown in Table (3-23) and the other parameters are as described in Table (3-21) and Table (3-22).

Table 3-23. User Specifiable Parameters for Equations 3-137 and 3-138			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	MUMINN.KLA	52.2	$cm^2(V*s)$
MOBILITY	MUMINP.KLA	44.9	$cm^2(V*s)$

The carrier-carrier scattering components, μ_{nc} and μ_{pc}, are given by:

$$\mu_{nc} = \frac{\mathbf{MUMINN.KLA} \times \mathbf{MUMAXN.KLA}}{\mathbf{MUMAXN.KLA} - \mathbf{MUMINN.KLA}} \left(\frac{300}{T_L}\right)^{0.5} \quad 3\text{-}139$$

$$\mu_{pc} = \frac{\mathbf{MUMINP.KLA} \times \mathbf{MUMAXP.KLA}}{\mathbf{MUMAXP.KLA} - \mathbf{MUMINP.KLA}} \left(\frac{300}{T_L}\right)^{0.5} \quad 3\text{-}140$$

The parameters of equations 3-135 and 3-136 N_{nsc} and N_{psc}, are given by:

$$N_{nsc} = N_D + N_A + p \quad 3\text{-}141$$

$$N_{psc} = N_D + N_A + n \quad 3\text{-}142$$

where N_D is the donor concentration in cm^{-3}, N_A is the acceptor concentration in cm^{-3}, n is the electron concentration in cm^{-3} and p is the hole concentration in cm^{-3}.

The parameters of equations 3-135 and 3-136, $N_{nsc,eff}$ and $N_{psc,eff}$ are given by:

$$N_{nsc,\,eff} = N_D + G(P_n)N_A + \left(\frac{p}{F(P_n)}\right) \quad 3\text{-}143$$

$$N_{psc,\,eff} = N_A + G(P_p)N_D + \left(\frac{n}{F(P_p)}\right) \quad 3\text{-}144$$

where N_D is the donor concentration in cm^{-3}, N_A is the acceptor concentration in cm^{-3} and n is the hole electron concentration in cm^{-3} and p is the hole concentration in cm^{-3}. The two functions G(P) and F(P) are functions of the screening factors P_n and P_p for electrons and holes. The function, G(P) in Equations 3-143 amd 3-145 are given by;

$$G(P_n) = 1 - \frac{\mathbf{S1.KLA}}{\left[\mathbf{S2.KLA} + P_{BH,\,n}\left(\frac{m_\circ}{m_e}\frac{T_L}{300}\right)^{\mathbf{S4.KLA}}\right]^{\mathbf{S3.KLA}}} + \frac{\mathbf{S5.KLA}}{\left[P_{BH,\,n}\left(\frac{m_\circ}{m_e}\frac{300}{T_L}\right)^{\mathbf{S7.KLA}}\right]^{\mathbf{S6.KLA}}} \quad 3\text{-}145$$

$$G(P_p) = 1 - \frac{\mathbf{S1.KLA}}{\left[\mathbf{S2.KLA} + P_{BH,p}\left(\frac{m_\circ}{m_h}\frac{T_L}{300}\right)^{\mathbf{S4.KLA}}\right]^{\mathbf{S3.KLA}}} + \frac{\mathbf{S5.KLA}}{\left[P_{BH,p}\left(\frac{m_\circ}{m_h}\frac{300}{T_L}\right)^{\mathbf{S7.KLA}}\right]^{\mathbf{S6.KLA}}} \quad 3\text{-}146$$

where T_L is the temperature in degrees Kelvin, m_e and m_h are the electron and hole masses and the parameters S1.KLA through S7.KLA are user specifiable model parameters as shown in Table 3-24.

.

Table 3-24. User Specifiable Parameters for Equations 3-145 and 3-146			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	S1.KLA	0.89233	
MOBILITY	S2.KLA	0.41372	
MOBILITY	S3.KLA	0.19778	
MOBILITY	S4.KLA	0.28227	
MOBILITY	S5.KLA	0.005978	
MOBILITY	S6.KLA	1.80618	
MOBILITY	S7.KLA	0.72169	

The functions $F(P_n)$ and $F(P_p)$ in equations 3-143 and 3-144 are given by:

$$F(P_n) = \frac{\mathbf{R1.KLA}\ P_n^{\mathbf{R6.KLA}} + \mathbf{R2.KLA} + \mathbf{R3.KLA}\ \frac{m_e}{m_h}}{P_n^{\mathbf{R6.KLA}} + \mathbf{R4.KLA} + \mathbf{R5.KLA}\ \frac{m_e}{m_h}} \quad 3\text{-}147$$

$$F(P_p) = \frac{\mathbf{R1.KLA}\ P_p^{\mathbf{R6.KLA}} + \mathbf{R2.KLA} + \mathbf{R3.KLA}\ \frac{m_h}{m_e}}{P_p^{\mathbf{R6.KLA}} + \mathbf{R4.KLA} + \mathbf{R5.KLA}\ \frac{m_h}{m_e}} \quad 3\text{-}148$$

where the parameters R1.KLA through R6.KLA are user specifiable model parameters as shown in Table 3-25.

Table 3-25. User Specifiable Parameters for Equations 3-147 and 3-148.			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	R1.KLA	0.7463	
MOBILITY	R2.KLA	2.2999	
MOBILITY	R3.KLA	6.5502	

Table 3-25. User Specifiable Parameters for Equations 3-147 and 3-148.			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	R4.KLA	2.3670	
MOBILITY	R5.KLA	-0.8552	
MOBILITY	R6.KLA	0.6478	

The screening parameters P_n and P_p used in the previous equations are given by:

$$P_n = \left[\frac{FCW.KLA}{P_{CW,n}} + \frac{FBH.KLA}{P_{BH,n}}\right]^{-1} \quad 3\text{-}149$$

$$P_p = \left[\frac{FCW.KLA}{P_{CW,p}} + \frac{FBH.KLA}{P_{BH,p}}\right]^{-1} \quad 3\text{-}150$$

where the parameters f_{CW} and f_{BH} are user specifiable model parameters as shown in Table 3-26.

Table 3-26. User Specifiable Parameters for Equations 3-149 and 3-150			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	FCW.KLA	2.459	
MOBILITY	FBH.KLA	3.828	

The function $P_{BH,n}$ and $P_{BH,p}$ are given by the following equations:

$$P_{BH,n} = \frac{1.36\times10^{20}}{n}\left(\frac{m_e}{m_0}\right)\left(\frac{T_L}{300}\right)^2 \quad 3\text{-}151$$

$$P_{BH,p} = \frac{1.36\times10^{20}}{p}\left(\frac{m_h}{m_0}\right)\left(\frac{T_L}{300}\right)^2 \quad 3\text{-}152$$

where T_L is the temperature in degrees Kelvin, m_e/m_0 and m_h/m_0 are the normalized carrier effective masses, and n and p are the electron and hole concentrations in cm^{-3}.

The functions, $P_{CW,n}$ and $P_{CW,p}$, of Eqs. 3-149 and 3-150 are given by:

$$P_{CW,n} = 3.97\times10^{13}\left\{\frac{1}{Z_n^3 N_D}\left(\frac{T_L}{300}\right)^3\right\} \quad 3\text{-}153$$

$$P_{CW,p} = 3.97\times10^{13}\left\{\frac{1}{Z_p^3 N_A}\left(\frac{T_L}{300}\right)^3\right\} \quad 3\text{-}154$$

where N_D and N_A are the donor and acceptor concentrations in cm^{-3}, T_L is the temperature in degrees Kelvin, and Z_n and Z_p are clustering functions given by:

$$Z_n = 1 + \frac{1}{\mathbf{CD.KLA} + \left(\frac{\mathbf{NREFD.KLA}}{N_D}\right)^2} \quad \text{3-155}$$

$$Z_p = 1 + \frac{1}{\mathbf{CA.KLA} + \left(\frac{\mathbf{NREFA.KLA}}{N_A}\right)^2} \quad \text{3-156}$$

where N_D and N_A are the donor and acceptor concentrations in cm^{-3} and CD.KLA, CA.KLA, NREFD.KLA and NREFA.KLA are user definable parameters as given in Table 3-27.

Table 3-27. User Specifiable Parameters for Equation 3-155 and 3-156

Statement	Parameter	Default	Units
MOBILITY	CD.KLA	0.21	
MOBILITY	CA.KLA	0.50	
MOBILITY	NREFD.KLA	4.0e20	cm^3
MOBILITY	NREFA.KLA	7.2e20	cm^3

Note: When the Klaassen low field mobility is used it should be remembered that it has been calibrated to work along with Klaassen's models for bandgap narrowing, AUGER recombination and SRH recombination. These models have been described in the carrier recombination section of this manual.

Inversion Layer Mobility Models

Overview

To obtain accurate results for MOSFET simulations, it is necessary to account for the mobility degradation that occurs inside inversion layers. The degradation normally occurs as a result of the substantially higher surface scattering near the semiconductor to insulator interface.

This effect is handled within ATLAS by three distinct methods:

- a surface degradation model SURFMOB
- a transverse electric field model SHIRAHATA
- specific inversion layer mobility models CVT, YAMAGUCHI and TASCH

The CVT, YAMAGUCHI and TASCH models are designed as stand-alone models which incorporate all the effects required for simulating the carrier mobility.

The Lombardi CVT Model

The inversion layer model from Lombardi [108] is selected by setting CVT on the MODEL statement. This model overrides any other mobility models which may be specified on the MODELS statement. In the CVT model, the transverse field, doping dependent and temperature dependent parts of the mobility are given by three components that are combined using Mathiessen's rule. These components are μ_{AC}, μ_{sr} and μ_b and are combined using Mathiessen's rule as follows:

$$\mu_T^{-1} = \mu_{AC}^{-1} + \mu_b^{-1} + \mu_{sr}^{-1} \quad \text{3-157}$$

The first component, μ_{AC}, is the surface mobility limited by scattering with acoustic phonons:

$$\mu_{AC,\,n} = \frac{\mathbf{BN.CVT}}{E_\perp} + \frac{\mathbf{CN.CVT}\ N^{\mathbf{TAU.CVT}}}{T_L E_\perp^{1/3}} \quad \text{3-158}$$

$$\mu_{AC,\,p} = \frac{\mathbf{BP.CVT}}{E_\perp} + \frac{\mathbf{CP.CVT}\ N^{\mathbf{TAUP.CVT}}}{T_L E_\perp^{1/3}} \quad \text{3-159}$$

where T_L is the temperature, $E_\perp$ is the perpendicular electric field, N is the total doping concentration. The equation parameters BN.CVT, BP.CVT, CN.CVT, CP.CVT, TAUN.CVT, and TAUP.CVT may be user defined on the MOBILITY statement and have the defaults shown in Table 3-28.

The second component, μ_{sr}, is the surface roughness factor and is given by:

$$\mu_{sr} = \frac{\mathbf{DELN.CVT}}{E_\perp^2} \quad \text{3-160}$$

$$\mu_{sr} = \frac{\mathbf{DELP.CVT}}{E_\perp^2} \quad \text{3-161}$$

The equation parameters DELN.CVT and DELP.CVT may be user defined on the MOBILITY statement and have the defaults shown in Table 3-28.

The third mobility component, μ_b, is the mobility limited by scattering with optical intervalley phonons. This component is given by:

$$\mu_{b,\,n} = \mathbf{MU0N.CVT}\exp\left(\frac{-\mathbf{PCN.CVT}}{N}\right) + \frac{\left[\mathbf{MUMAXN.CVT}\left(\frac{T_L}{300}\right)^{-\mathbf{GAMN.CVT}} - \mathbf{MU0N.CVT}\right]}{1 + \left(\frac{N}{\mathbf{CRN.CVT}}\right)^{\mathbf{ALPHN.CVT}}}$$

$$- \frac{\mathbf{MU1N.CVT}}{1 + \left(\frac{\mathbf{CSN.CVT}}{N}\right)^{\mathbf{BETAN.CVT}}} \quad \text{3-162}$$

$$\mu_{b,p} = \mathbf{MU0P.CVT}\exp\left(\frac{-\mathbf{PCP.CVT}}{N}\right) + \frac{\left[\mathbf{MUMAXP.CVT}\left(\frac{T_L}{300}\right)^{-\mathbf{GAMP.CVT}} - \mathbf{MU0P.CVT}\right]}{1+\left(\frac{N}{\mathbf{CRP.CVT}}\right)^{\mathbf{ALPHP.CVT}}}$$

$$- \frac{\mathbf{MU1P.CVT}}{1+\left(\frac{\mathbf{CSP.CVT}}{N}\right)^{\mathbf{BETAP.CVT}}}$$

3-163

Where N is the total density of impurities and T_L is the temperature in degrees Kelvin.

Table 3-28. User Specifiable Parameters for Equations 3-158 to 3-163

Statement	Parameter	Default	Units
MOBILITY	BN.CVT	$4.75*10^{7}$	cm/(s)
MOBILITY	BP.CVT	$9.925*10^{6}$	cm/(s)
MOBILITY	CN.CVT	$1.74*10^{5}$	
MOBILITY	CP.CVT	$8.842*10^{5}$	
MOBILITY	TAUN.CVT	0.125	unitless
MOBILITY	TAUP.CVT	0.0317	unitless
MOBILITY	GAMN.CVT	2.5	unitless
MOBILITY	GAMP.CVT	2.2	unitless
MOBILITY	MU0N.CVT	52.2	$cm^2/(V\cdot s)$
MOBILITY	MU0P.CVT	44.9	$c^{m2}/(V\cdot s)$
MOBILITY	MU1N.CVT	43.4	$cm^2/(V\cdot s)$
MOBILITY	MU1P.CVT	29.0	cm2/(V·s)
MOBILITY	MUMAXN.CVT	1417.0	$cm^2/(V\cdot s)$
MOBILITY	MUMAXP.CVT	470.5	$cm^2/(V\cdot s)$
MOBILITY	CRN.CVT	$9.68*10^{16}$	cm^{-3}
MOBILITY	CRP.CVT	$2.23*10^{17}$	cm^{-3}
MOBILITY	CSN.CVT	$3.43*10^{20}$	cm^{-3}
MOBILITY	CSP.CVT	$6.10*10^{20}$	cm^{-3}
MOBILITY	ALPHN.CVT	0.680	

Table 3-28. User Specifiable Parameters for Equations 3-158 to 3-163			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	ALPHP.CVT	0.71	
MOBILITY	BETAN.CVT	2.00	
MOBILITY	BETAP.CVT	2.00	
MOBILITY	PCN.CVT	0.0	cm^{-3}
MOBILITY	PCP.CVT	$0.23*10^{16}$	cm^{-3}
MOBILITY	DELN.CVT	$5.82*10^{14}$	V/s
MOBILITY	DELP.CVT	$2.0546*10^{14}$	V^2/s

Note: The CVT model when activated will also, by default, apply the parallel electric field mobility model which is described in a later section of this chapter. In this model the low field mobility is supplied from the CVT model.

The Yamaguchi Model

The Yamaguchi model is selected by setting YAMAGUCHI on the MODELS statement. This model overrides any mobility model specifications other than the CVT model. The model consists of calculating the low field, doping dependent mobility. Surface degradation is then accounted for based upon the transverse electric field before including the parallel electric field dependence.

The low field part of the Yamaguchi model is given as follows:

$$\mu_{n0} = \mathbf{MULN.YAMA}\left[1 + \frac{N_i}{\frac{N_i}{\mathbf{SN.YAMA}} + \mathbf{NREFP.YAMA}}\right]^{-\frac{1}{2}} \quad 3\text{-}164$$

$$\mu_{p0} = \mathbf{MULP.YAMA}\left[1 + \frac{N_i}{\frac{N_i}{\mathbf{SP.YAMA}} + \mathbf{NREFP.YAMA}}\right]^{-\frac{1}{2}} \quad 3\text{-}165$$

where N_i is the net impurity concentration. The equation parameters MULN.YAMA, MULP.YAMA, SN.YAMA, SP.YAMA, NREFP.YAMA and NREFP.YAMA are user definable on the MOBILITY statement and have the defaults shown in Table 3-29.

The transverse electric field dependence is accounted for as follows:

$$\mu_{s,n} = \mu_{n0}(1 + \mathbf{ASN.YAMA}\ E_{\perp})^{-\frac{1}{2}} \quad 3\text{-}166$$

$$\mu_{s,p} = \mu_{p0}(1 + \mathbf{ASP.YAMA}\ E_{\perp})^{-\frac{1}{2}} \quad 3\text{-}167$$

where $E_{\perp}$ is the perpendicular electric field and the equation parameters ASN.YAMA and ASP.YAMA are user definable on the MOBILITY statement and have the defaults shown in Table 3-29.

The final calculation of mobility takes into acount the parallel electric field dependence which takes the form:

$$\mu_n = \mu_{s,n}\left[1 + \left(\frac{\mu_{s,n}E}{\mathbf{ULN.YAMA}}\right)^2\left(\frac{\mathbf{GN.YAMA} + \mu_{s,n}E}{\mathbf{ULN.YAMA}}\right)^{-1} + \left(\frac{\mu_{s,n}E}{\mathbf{VSN.YAMA}}\right)^2\right]^{-\frac{1}{2}} \quad 3\text{-}168$$

$$\mu_p = \mu_{s,p}\left[1 + \left(\frac{\mu_{s,p}E}{\mathbf{ULP.YAMA}}\right)^2\left(\frac{\mathbf{GP.YAMA} + \mu_{s,p}E}{\mathbf{ULP.YAMA}}\right)^{-1} + \left(\frac{\mu_{s,p}E}{\mathbf{VSP.YAMA}}\right)^2\right]^{-\frac{1}{2}} \quad 3\text{-}169$$

where E is the parallel electric field and the equation parameters ULN.YAMA, ULP.YAMA, VSN.YAMA, VSP.YAMA, GN.YAMA and GP.YAMA are user definable on the MOBILITY statement and have the defaults shown in Table 3-29.

Table 3-29. User Specifiable Parameters for Equations 3-164 to 3-169.

Statement	Parameter	Default	Units
MOBILITY	SN.YAMA	350.0	
MOBILITY	SP.YAMA	81.0	
MOBILITY	NREFP.YAMA	$3.0*10^{16}$	cm^{-3}
MOBILITY	NREFP.YAMA	$4.0*10^{16}$	cm^{-3}
MOBILITY	MULN.YAMA	1400.0	$cm^2/(V \cdot s)$
MOBILITY	MULP.YAMA	480.0	$cm^2/(V \cdot s)$
MOBILITY	ASN.YAMA	$1.54*10^{-5}$	cm/V
MOBILITY	ASP.YAMA	$5.35*10^{-5}$	cm/V
MOBILITY	VSN.YAMA	$1.036*10^{7}$	cm/s
MOBILITY	VSP.YAMA	$1.200*10^{7}$	cm/s
MOBILITY	ULN.YAMA	$4.9*10^{6}$	cm/s
MOBILITY	ULP.YAMA	$2.92*10^{6}$	cm/s
MOBILITY	GN.YAMA	8.8	cm/s
MOBILITY	GP.YAMA	1.6	cm/s

The Tasch Model

S-PISCES includes an improved local field-dependent mobility model. This model, which was originally derived and published by Tasch et. al, has been designed explicitly for MOSFETs [7,9]. It defines the mobility as a function of the perpendicular and parallel electric fields, the interface charge, the lattice temperature and the doping concentration.This model is activated by the parameter TASCH on the MODELS statement. This mobility model is given by the following expressions:

$$\mu_n = \Gamma_n + (E_{perp} - E_0)\frac{d\Gamma_n}{dE_{per}} \quad 3\text{-}170$$

$$\mu_p = \Gamma_p + (E_{perp} - E_0)\frac{d\Gamma_p}{dE_{per}} \quad 3\text{-}171$$

where Eperp is the transverse electric field and E_0 is the transverse electric field at the edge of the inversion layer. The functions $\Gamma_{n,p}$ are defined as:

$$\Gamma_n = \frac{\mu_{eff,\,n}}{\left(1 + \left(\frac{\mu_{eff,\,n} E_{\parallel}}{\mathbf{VSATN}}\right)^{\mathbf{BETAN}}\right)^{1/\mathbf{BETAN}}} \quad 3\text{-}172$$

$$\Gamma_p = \frac{\mu_{eff,\,p}}{\left(1 + \left(\frac{\mu_{eff,\,p} E_{\parallel}}{\mathbf{VSATP}}\right)^{\mathbf{BETAP}}\right)^{1/\mathbf{BETAP}}} \quad 3\text{-}173$$

The carrier mobilities $\mu_{eff,n}$ and $\mu_{eff,p}$ are defined by three components μ_{ph}, μ_{sr} and μ_c which are combined by Mathiessen's rule according to:

$$\mu_{eff} = \left[\frac{1}{\mu_{ph}} + \frac{1}{\mu_{sr}} + \frac{1}{\mu_c}\right]^{-1} \quad 3\text{-}174$$

The term μ_{ph} takes account of the degradation in mobility due to acoustic phonon scattering through the expressions:

$$\mu_{ph,\,n}^{-1} = \left(\mathbf{MUBN.TAS}\left(\frac{T_L}{300}\right)^{-\mathbf{TMUBN.TAS}}\right)^{-1} + \left(Z_n/\mathbf{DN.TAS}\; Y_n\left(\frac{T_L}{300}\right)^{1/2}\right)^{-1} \quad 3\text{-}175$$

$$\mu_{ph,\,p}^{-1} = \left(\mathbf{MUBP.TAS}\left(\frac{T_L}{300}\right)^{-\mathbf{TMUBP.TAS}}\right)^{-1} + \left(Z_p/\mathbf{DP.TAS}\; Y_p\left(\frac{T_L}{300}\right)^{1/2}\right)^{-1} \quad 3\text{-}176$$

where the functions $Z_{n,p}$ and $Y_{n,p}$ are defined as:

$$Z_n = \mathbf{Z11N.TAS}\left(\frac{T_L}{300}\right)E_{eff,\,n}^{-1} + \mathbf{Z22N.TAS}\; E_{eff,\,n}^{-\frac{1}{3}} \quad 3\text{-}177$$

$$Z_p = \mathbf{Z11P.TAS}\left(\frac{T_L}{300}\right)E_{eff,\,p}^{-1} + \mathbf{Z22P.TAS}\; E_{eff,\,p}^{-\frac{1}{3}} \quad 3\text{-}178$$

$$E_{eff,\,n} = \frac{(E_{perp} + (\mathbf{RN.TAS} - 1)\cdot E_0)}{\mathbf{RN.TAS}} \quad 3\text{-}179$$

$$E_{eff,\,p} = \frac{(E_{perp} + (\mathbf{RP.TAS} - 1)\cdot E_0)}{\mathbf{RP.TAS}} \quad 3\text{-}180$$

$$Y_n = \mathbf{P1N.TAS}\left(\frac{T_L}{300}\right)^{\mathbf{B1N.TAS}} + \mathbf{P2N.TAS}\; n^{-\mathbf{B2N.TAS}}\left(\frac{T_L}{300}\right)^{-1} N_f \quad 3\text{-}181$$

$$Y_p = \mathbf{P1P.TAS}\left(\frac{T_L}{300}\right)^{\mathbf{B1P.TAS}} + \mathbf{P2P.TAS}\; p^{-\mathbf{B2P.TAS}}\left(\frac{T_L}{300}\right)^{-1} N_f \quad 3\text{-}182$$

Mobility degradation due to surface roughness is accounted for by the term μ_{sr} which is calculated according to:

$$\mu_{sr,\,n} = (\mathbf{ESRN.TAS}\; E_{eff,\,n})^{-\mathbf{BETAN.TAS}} \quad 3\text{-}183$$

$$\mu_{sr,\,p} = (\mathbf{ESRP.TAS}\; E_{eff,\,p})^{-\mathbf{BETAP.TAS}} \quad 3\text{-}184$$

The final term μ_C models Coulombic scattering with the expressions:

$$\mu_{C,\,n} = \frac{\mathbf{N2N.TAS}\cdot\left(\frac{T_L}{300}\right)^{1.5}}{N_A \ln(1+\gamma_{BH,\,n})\cdot\frac{\gamma_{BH,\,n}}{(1+\gamma_{BH,\,n})}} \quad 3\text{-}185$$

$$\mu_{C,\,p} = \frac{\mathbf{N2P.TAS}\cdot\left(\frac{T_L}{300}\right)^{1.5}}{N_D \ln(1+\gamma_{BH,\,p})\cdot\frac{\gamma_{BH,\,p}}{(1+\gamma_{BH,\,p})}} \quad 3\text{-}186$$

where

$$\gamma_{BH,\,n} = \frac{\mathbf{N1N.TAS}}{n}\cdot\left(\frac{T_L}{300}\right)^{\mathbf{ALPHAN.TAS}} \quad 3\text{-}187$$

$$\gamma_{BH,\,p} = \frac{\mathbf{N1P.TAS}}{p}\cdot\left(\frac{T_L}{300}\right)^{\mathbf{ALPHAP.TAS}} \quad 3\text{-}188$$

In the above equations T_L is the lattice temperature in degrees Kelvin, N_f is the fixed interface charge at the gate dielectric-silicon interface (cm^{-2}), N_A is the channel acceptor doping concentration in cm^{-3}, N_D is the channel donor doping concentration in cm^{-3}, n and p are the electron and hole

concentrations per unit volume in the inversion layer (cm^{-3}). The default parameters within each equation may be defined on the MOBILITY statement. The default parameters are shown in Table 3-30.

Table 3-30. Parameters for Equations 3-172 through 3-188.

Statement	Parameter	Default	Units
MOBILITY	RN.TAS	2	
MOBILITY	RP.TAS	3	
MOBILITY	BETAN	2	
MOBILITY	BETAP	1	
MOBILITY	BETAN	2	
MOBILITY	BETAP	1	
MOBILITY	BETAN	1.5	
MOBILITY	BETAN		
MOBILITY	MUBN.TAS	1150	
MOBILITY	MUBP.TAS	270	
MOBILITY	TMUBN.TAS	2.5	
MOBILITY	TMUBP.TAS	1.4	
MOBILITY	DN.TAS	$3.2*10^{-9}$	
MOBILITY	DP.TAS	$2.35*10^{-9}$	
MOBILITY	P1N.TAS	0.09	
MOBILITY	P1P.TAS	0.334	
MOBILITY	B1N.TAS	1.75	
MOBILITY	B1P.TAS	1.5	
MOBILITY	P2N.TAS	$4.53*10^{-8}$	
MOBILITY	PEP.TAS	$3.14*10^{-7}$	
MOBILITY	B2N.TAS	-0.25	
MOBILITY	B2P.TAS	-0.3	
MOBILITY	Z11N.TAS	0.0388	
MOBILITY	Z11P.TAS	0.039	
MOBILITY	Z22N.TAS	$1.73*10^{-5}$	
MOBILITY	Z22P.TAS	$1.51*10^{-5}$	
MOBILITY	ESRN.TAS	$2.449*10^{7}$	

Table 3-30. Parameters for Equations 3-172 through 3-188.			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	ESRP.TAS	$10*10^{8}$	
MOBILITY	BETAN.TAS	2	
MOBILITY	BETAP.TAS	1	
MOBILITY	N2N.TAS	$1.1*10^{21}$	
MOBILITY	N2P.TAS	$1.4*10^{18}$	
MOBILITY	N1N.TAS	$2*10^{19}$	
MOBILITY	N1P.TAS	$8.4*10^{16}$	
MOBILITY	ALPHAN.TAS	2	
MOBILITY	ALPHAP.TAS	3.4	

Perpendicular Electric Field-Dependent Mobility Models

The Watt Model

A surface mobility model derived by J.T.Watt [5] is available in ATLAS. This mobility model is activated when the parameter SURFMOB is specified on the MODELS statement. The default model parameters are tuned to describe the measured mobilities in silicon at 300K. Users may modify model parameters using the MOBILITY statement.

1. The Watt model takes into consideration the following primary scattering mechanisms in the inversion layer:
2. Phonon scattering which results primarily from the interaction between two-dimensional inversion layer carriers and bulk phonons.
3. Surface roughness scattering caused by the interaction between inversion layer carriers and deviations from ideal planarity at the interface.

Charged impurity scattering caused by the interaction between inversion layer carriers and ions located in the oxide, at the interface, or in the bulk

The phonon and surface roughness components are functions of effective electric field. The charged impurity component is a function of the channel doping density.

The effective mobilities for electrons and holes are given by Equations 3-189 and 3-190.

$$\frac{1}{\mu_{eff,\,n}} = \frac{1}{\mathbf{MREF1N.WATT}}\left(\frac{1}{E_{eff,\,n}}\right)^{\mathbf{AL1N.WATT}} + \frac{1}{\mathbf{MREF2N.WATT}}\left(\frac{1}{E_{eff,\,n}}\right)^{\mathbf{AL2N.WATT}} + \frac{1}{\mathbf{MREF3N.WATT}}\left(\frac{1}{N_B}\right)^{-1}\left(\frac{1}{N_i}\right)^{\mathbf{AL3N.WATT}} \quad \text{3-189}$$

$$\frac{1}{\mu_{eff,\,p}} = \frac{1}{\mathbf{MREF1P.WATT}}\left(\frac{1}{E_{eff,\,p}}\right)^{\mathbf{AL1P.WATT}} + \frac{1}{\mathbf{MREF2P.WATT}}\left(\frac{1}{E_{eff,\,p}}\right)^{\mathbf{AL2P.WATT}} + \frac{1}{\mathbf{MREF3P.WATT}}\left(\frac{1}{N_B}\right)^{-1}\left(\frac{1}{N_i}\right)^{\mathbf{AL3N.WATT}} \quad 3\text{-}190$$

Where N_B is the surface trapped charge density, N_i is the inversion layer charge density and E_{eff} is the effective electric field given by:

$$E_{eff,\,n} = E_{\perp} + \mathbf{ETAN.WATT}(E_0 - E_{\perp}) \quad 3\text{-}191$$

$$E_{eff,\,p} = E_{\perp} + \mathbf{ETAP.WATT}(E_0 - E_{\perp}) \quad 3\text{-}192$$

where $E_{\perp}$ is the electric field perpendicular to the current flow and E_o is the perpendicular electric field at the insulator-semiconductor interface. The equation parameters and their defaults are listed in Table 3-31.

In Equations 3-191 and 3-192. E_{eff} represents the effective electric field. The terms on the right side of Equation 3-189 describe (in order) the three scattering mechanisms previously discussed. Each component contains two constants: a pre-exponential factor and the exponent of the principal independent parameter. The charge impurity scattering component is assumed to be inversely proportional to doping density.

The expression for effective mobility contains a number of normalizing constants. These normalizing constants are included to allow easy comparison of constants. The first two terms in this mobility model are dependent on E_{eff} and represent the universal mobility-field relationship. The third term accounts for deviation from the universal relationship resulting from charged impurity scattering.

Table 3-31. User Specifiable Parameters for Equations 3-189 - 3-192

Statement	Parameter	Default	Units
MOBILITY	ETAN.WATT	0.50	
MOBILITY	ETAP.WATT	0.33	
MOBILITY	MREF1N.WATT	481.0	cm2/(V·s)
MOBILITY	MREF1P.WATT	92.8	cm2/(V·s)
MOBILITY	MREF2N.WATT	591.0	cm2/(V·s)
MOBILITY	MREF2P.WATT	124.0	cm2/(V·s)
MOBILITY	MREF3N.WATT	1270.0	cm2/(V·s)
MOBILITY	MREF3P.WATT	534.0	cm2/(V·s)
MOBILITY	AL1N.WATT	-0.16	
MOBILITY	AL1P.WATT	-0.296	

Table 3-31. User Specifiable Parameters for Equations 3-189 - 3-192			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	AL2N.WATT	-2.17	
MOBILITY	AL2P.WATT	-1.62	
MOBILITY	AL3N.WATT	1.07	
MOBILITY	AL3P.WATT	1.02	

Modifications to Watt's Model

By default the Watt mobility model is a surface model that applies, only to those grid points on the silicon/ oxide interface. A modification has been added that now applies the Watt model to points below the interface. This extension to the Watt model is enabled using the MOD.WATT.N and MOD.WATT.P parameters of the MOBILITY statement.

The distance over which the model is applied can be controlled by using the YMAXN.WATT and the YMAXP.WATT parameters of the MOBILITY statement. These parameters specify the maximum value of the y-coordinate over which the model is applies below the interface for electrons and holes respectively.

The XMINN.WATT, XMINP.WATT, XMAXN.WATT and XMAXP.WATT of the MOBILITY statement can be used to limit the range of the model in the x-direction, to prevent the model from applying to the source and drain regions. The MIN.SURF parameter of the MODELS statement can also be used for this purpose. When enabled, the MIN.SURF parameter will ensure that the Watt model will only apply to minority regions.

The logical parameters EXP.WATT.N and EXP.WATT.P of the MOBILITY statement can also be used to enable a additional modification to the Watt model. When these parameters are enabled the effective normal electric field becomes a function of the depth beneath the silicon/oxide interface according to:

$$E_{\perp, n} = E_y \exp \frac{-(y - y_{int})}{\mathbf{YCHARN.WATT}} \qquad 3\text{-}193$$

$$E_{\perp, p} = E_y \exp \frac{-(y - y_{int})}{\mathbf{YCHARP.WATT}} \qquad 3\text{-}194$$

where $E_\perp$ is the perpendicular electric field, E_y is the perpendicular electric field at the interface, y is the local y-coordinate and y_{int} is the y-coordinate of the silicon/oxide interface. The equation parameters YCHARN.WATT and YCHARP.WATT are user definable on the MOBILITY statement.

Shirahata's Mobility Model

The Shirahata mobility model [111] is general purpose MOS mobility model that takes into account screening effects in the inversion layer as well as improved perpendicular field dependence for thin gate oxides. In the original paper, the authors, present the model as a combination of portions of Klaassen's model for low field mobility contributions and an empirically fit expression for the perpendicular field dependent mobility in the inversion layer. In our implementation, we simply fit our more complete implementation of Klaassen's model beneath the same perpendicular field model used in the original paper.

The Shirahata model is enabled by the SHI parameter of the MODELS statement or can be enabled individually for electrons and holes using the SHI.N and SHI.P parameters of the MOBILITY statement. If a low field mobility model is also chosen such as KLA then Mathiessens rule is used to combine the two models. The following paragraphs describe the analytic form of the Shirahata mobility model.

The Shirahata model for electrons and holes are given by:

$$\mu_n = \frac{\mathbf{MU0N.SHI}}{\left[1 + \frac{|E_\perp|}{\mathbf{E1N.SHI}}\right]^{\mathbf{P1N.SHI}} + \left[\frac{|E_\perp|}{\mathbf{E2N.SHI}}\right]^{\mathbf{P2N.SHI}}} \quad \text{3-195}$$

$$\mu_p = \frac{\mathbf{MU0P.SHI}}{\left[1 + \frac{|E_\perp|}{\mathbf{E1P.SHI}}\right]^{\mathbf{P1P.SHI}} + \left[\frac{|E_\perp|}{\mathbf{E2P.SHI}}\right]^{\mathbf{P2P.SHI}}} \quad \text{3-196}$$

where $E_\perp$ is the perpendicular electric and the equation parameters MU0N.SHI, MU0P.SHI, E1N.SHI. E1P.SHI, E2N.SHI, E2P.SHI, P1N.SHI, P1P.SHI, P2N.SHI and P2P.SHI are user definable on the MOBILITY statement and have the defaults shown in Table 3-32.

Table 3-32. User Specifiable Parameters for Equations 3-195 and 3-196.

Statement	Parameter	Default	Units
MOBILITY	MU0N.SHI	1400.0	$cm^2/(V \cdot s)$
MOBILITY	MU0P.SHI	500.0	$cm^2/(V \cdot s)$
MOBILITY	E1N.SHI	$8.9x10^3$	V/cm
MOBILITY	E1P.SHI0	$8.0x10^3$	V/cm
MOBILITY	E2N.SHI	$1.22x10^6$	V/cm
MOBILITY	E2P.SHI	$3.9x10^5$	V/cm
MOBILITY	P1N.SHI	0.28	
MOBILITY	P1P.SHI	0.3	
MOBILITY	P2N.SHI	2.9	
MOBILITY	P2P.SHI	1.0	

Note: If the maximum low field mobility has been user defined then it is important to also define this value inside the Shirahata model with the parametes MU0N.SHI and MU0P.SHI on the MOBILITY statement.

Parallel Electric Field-Dependent Mobility

As carriers are accelerated in an electric field their velocity will begin to saturate at high enough electric fields. This effect has to be accounted for by a reduction of the effective mobility since the magnitude of the drift velocity is the product of the mobility and the electric field component in the direction of the current flow. The following Caughey and Thomas expression [22] is used to implement a field-dependent mobility that provides a smooth transition between low-field and high field behavior:

$$\mu_n(E) = \mu_{n0}\left[\frac{1}{1+\left(\frac{\mu_{n0}\,E}{\mathbf{VSATN}}\right)^{\mathbf{BETAN}}}\right]^{\frac{1}{\mathbf{BETAN}}} \quad \text{3-197}$$

$$\mu_p(E) = \mu_{p0}\left[\frac{1}{1+\left(\frac{\mu_{p0}\,E}{\mathbf{VSATP}}\right)^{\mathbf{BETAP}}}\right]^{\frac{1}{\mathbf{BETAP}}} \quad \text{3-198}$$

where E is the parallel electric field and μ_{n0} and μ_{p0} are the low field electron and hole mobilities respectively. The low field mobilities are either set explicitly on the MOBILITY statement or calculated by one of the low field mobility models. The model parameters BETAN and BETAP are user definable on the MOBILITY statement and have the default values shown in Table 3-35.

The saturation velocities are calculated by default from the temperature dependent model [13]:

$$\mathbf{VSATN} = \mathbf{VSATP} = \frac{2.4\times10^{7}}{1+0.8\exp\left(\frac{T_L}{600}\right)} \quad \text{3-199}$$

but can be set to constant values on the MOBILITY statement using the parameters VSATN and VSATP. In this case no temperature dependence is implemented. Specifying the FLDMOB parameter on the MODELS statement invokes the field-dependent mobility. FLDMOB should always be specified unless one of the inverson layer mobility models (which incorporate their own dependence on the parallel field) are specified.

It is possible to invoke a C-interpreter function for the saturation velocities. The parameter F.VSATN and F.VSATP of the MATERIAL statement may be set to provide the filenames of two text files containing the particular functions. These functions allow the temperature dependence to be included. See Appendix A for more details.

Table 3-33. User Definable Parameters in the Field-Dependent Mobility Model			
Statement	**Parameter**	**Default**	**Units**
MOBILITY	BETAN	2.0	
MOBILITY	BETAP	1.0	
MOBILITY	VSATN		cm/s
MOBILITY	VSATP		cm/s

Note: The above model, which was derived for the drift-diffusion approximation, ensures that velocity overshoot can not occur. To model velocity overshoot in silicon the energy balance model must be applied. This model follows the above implementation but with the electric field term replaced by a new "effective" field calculated from the carrier temperature - see the following section for more details.

Note: BLAZE includes a different field dependent mobility model that does simulate velocity overshoot in GaAs, this is described in the BLAZE chapter.

Carrier Temperature Dependent Mobility

The energy balance transport model allows the carrier mobility to be related to the carrier energy. This has been achieved through the homogeneous steady state energy balance relationship that pertains in the saturated velocity limit. This allows an effective electric field to be calculated which is the uniform electric field value which causes the carriers in an homogeneous sample to attain the same temperature as at the node point in the device. The effective electric fields $E_{eff,n}$ and $E_{eff,p}$ are calculated by solving the equations:

$$q\mu_n(E_{eff,n})E_{eff,n}^2 = \frac{3}{2}\frac{k(T_n - T_L)}{\mathbf{TAUREL.EL}} \quad \text{3-200}$$

$$q\mu_p(E_{eff,p})E_{eff,p}^2 = \frac{3}{2}\frac{k(T_p - T_L)}{\mathbf{TAUREL.HO}} \quad \text{3-201}$$

for $E_{eff,n}$ and $E_{eff,p}$. These equations are derived friom the energy balance equations by stripping out all spatially varying terms. The effective electric fields are then introducecd into the relevent field dependent mobility model.

Four different models have been implemented into the ATLAS energy balance model which can be chosen by the parameter EVSATMOD on the MODELS statement. These four models shall be described next.

Setting EVSATMOD=0 implements the default model for silicon, based upon the Caughey-Thomas field-dependent mobility model in Equation 3-197. The resultant relationship between the carrier mobility and the carrier temperature is of the form;

$$\mu_n = \frac{\mu_{n0}}{(1 + X_n^{\mathbf{BETAN}})^{\frac{1}{\mathbf{BETAN}}}} \quad 3\text{-}202$$

$$\mu_p = \frac{\mu_{p0}}{(1 + X_p^{\mathbf{BETAP}})^{\frac{1}{\mathbf{BETAP}}}} \quad 3\text{-}203$$

$$X_n^{\mathbf{BETAN}} = \frac{1}{2}(\alpha_n^{\mathbf{BETAN}} + \sqrt{\alpha_n^{2\mathbf{BETAN}}(T_n - T_L)^{2\mathbf{BETAN}} - 4\alpha^{\mathbf{BETAN}}(T_n - T_L)^{\mathbf{BETAN}}}) \quad 3\text{-}204$$

$$X_p^{\mathbf{BETAP}} = \frac{1}{2}(\alpha_p^{\mathbf{BETAP}} + \sqrt{\alpha_p^{2\mathbf{BETAP}}(T_p - T_L)^{2\mathbf{BETAP}} - 4\alpha^{\mathbf{BETAP}}(T_p - T_L)^{\mathbf{BETAP}}}) \quad 3\text{-}205$$

$$\alpha_n = \frac{3}{2}\frac{k_B\mu_{n0}}{q\mathbf{VSATN}^2\,\mathbf{TAUREL.EL}} \quad 3\text{-}206$$

$$\alpha_p = \frac{3}{2}\frac{k_B\mu}{q\mathbf{VSATP}^2\,\mathbf{TAUREL.HO}} \quad 3\text{-}207$$

where μ_{n0} and μ_{p0} are the low field carrier mobilities and VSATN and VSATP are the saturated velocities for electrons and holes. The parameters VSATN, VSATP, and BETAN and BETAP are user definable on the MOBILITY statement. The terms TAUREL.EL and TAUREL.HO are the energy relaxation times for electrons and holes and may be user defined on the MATERIAL statement.

Setting EVSATMOD=0 with the additional parameter MOBTEM.SIMPL allows a simplified form of the above model to be applied. This model has the form;

$$\mu_n = \frac{\mu_{n0}}{\sqrt{1 + \alpha_n^{\,2}(T_n - T_L)^2}} \quad 3\text{-}208$$

$$\mu_p = \frac{\mu_{p0}}{\sqrt{1 + \alpha_p^{\,2}(T_p - T_L)^2}} \quad 3\text{-}209$$

where μ_{n0} and μ_{p0} are again the low field carrier mobilities and $\alpha_{n,p}$ are as defined above.

Setting EVSATMOD=1 implements the GaAs carrier temperature dependent mobilty model. This model is described in the BLAZE chapter of this manual.

Setting EVSATMOD=2 will apply the simple velocity limiting model based upon the electric field. In other words the temperature dependent mobility is turned off and the standard electric field based mobility model is applied.

Note: If the CVT, YAMAGUCHI or TASCH mobility models are chosen in the MODEL statement then no energy dependence is applied. No energy dependence is included in any perpendicular electric field model such as SHIRAHATA or SURFMOB.

Table 3-34. User Specifiable Parameters for Equations 3-197 - 3-207

Statement	Parameter	Units
MOBILITY	MUN	$cm^2/(Vs)$
MOBILITY	MUP	$cm^2/(Vs)$
MATERIAL	VSATN	cm/s
MATERIAL	VSATP	cm/s

Mobility Model Summary

Model	Syntax	Notes
Concentration Dependent	CONMOB	Lookup table valid at 300K for Si and GaAs only. Uses simple power law temperature dependence.
Concentration and Temperature Dependent	ANALYTIC	Caughey-Thomas formula. Tuned for 77-450K.
Arora's Model	ARORA	Alternative to ANALYTIC for Si
Carrier-Carrier Scattering	CCSMOB	Dorkel-Leturq Model. Includes n, N and T dependence.. Important when carrier concentration is high (e.g., forward bias power devices).
Parallel Electric Field Dependence	FLDMOB	Si and GaAs models. Required to model any type of velocity satuation effect.
Tasch Model	TASCH	Includes transverse field dependence. Only for planar devices. Needs very fine grid.
Watt Model	WATT	Transverse field model applied to surface nodes only.
Klaassen Model	KLA	Includes N, T and n dependence. Applies separate mobility to majority and minority carriers. Recommended for bipolar devices
Shirahata Model	SHI	Includes N, $E_\perp$. An alternative surface mobility model that can be combined with KLA.
Modified Watt	MOD.WATT	Extension of WATT model to non-surface nodes. Applies constant $E_\perp$ effects. Best model for planar MOS devices
Lombardi (CVT) Model	CVT	Complete model including N, T, E// and $E_\perp$ effects. Good for non-planar devices.
Yamaguchi Model	YAMAGUCHI	Includes N, E// and $E_\perp$ effects. Only for 300K.

	CONMOB	BLDMOB	TFLDMB2	YAMAGUCHI	CVT	ARORA	ANALYTIC	CCSMOB	SURFACE	LATTICE H	E.BALANCE
CONMOB [CM]	—	OK	OK	YA	CV	AR	AN	CC	OK	OK	OK
FLDMOB [FM]	OK	—	TF[1]	YA	CV	OK	OK	OK	OK	OK	OK
TFLDMB2 [TF]	OK	TF[1]	—	YA	CV	OK	OK	TF	TF	OK	OK
YAMAGUCHI [YA]	YA	YA	YA	—	CV	YA	YA	YA	YA	NO	NO
CVT [CV]	CV	CV	CV	CV	—	CV	CV	CV	CV	OK	OK
ARORA [AR]	AR	OK	OK	YA	CV	—	AR	CC	OK	OK	OK
ANALYTIC [AN]	AN	OK	OK	YA	CV		—	CC	OK	OK	OK
CCSMOB [CC]	CC	OK	TF	YA	CV	CC	CC	—	OK	OK	OK
SURFMOB [SF]	OK	OK	TF	YA	CV	OK	OK	OK	—	OK	OK
LATTICE H [LH]	OK	OK	OK	NO	OK	OK	OK	OK	OK	—	OK
E.BALANCE [EB]	OK	OK	OK	NO	OK	OK	OK	OK	OK	OK	[2]

Key to Table Entries:

MODEL ABBREVIATION = The model that supercedes when a combination is specified. In some cases, but not all, a warning message is issued when a model is ignored.

OK = This combination is allowed

NO = This combination is not allowed

NOTES:

1 Uses internal model similar to FLDMOB

2 When models including a parallel electric field dependence are used with energy balance the electric field term is replaced by a function of carrier temperature.

Carrier Generation-Recombination Models

To put it simply carrier generation-recombination is the process through which the semiconductor material attempts to return to equilibrium after being disturbed from it. If we consider an homogeneously doped semiconductor with equilibrium concentrations n_0 an p_0 then at equilibrium a steady state balance exists according to:

$$n_0 = p_0 = n_i^2 \qquad 3\text{-}210$$

However, semiconductors are under continual excitation whereby n_0 and p_0 are disturbed from their equilibrium states. For instance, light shining on the surface of a p type semiconductor causes generation of electron-hole pairs, disturbing greatly the minority carrier concentration. A net recombination results which attempts to return the semiconductor to equilibrium. The processes responsible for generation-recombination are known to fall into six main categories:

- phonon transitions
- photon transitions
- Auger transitions
- surface recombination
- impact ionization
- tunneling

The following sections will describe the models implemented into ATLAS that attempt to simulate these six types of generation-recombination mechanisms.

Shockley-Read-Hall (SRH) Recombination

Phonon transitions occur in the presence of a trap (or defect) within the forbidden gap of the semiconductor. This is essentially a two step process, the theory of which was first derived by Shockley and Read [133] and then by Hall [132]. Shockley-Read-Hall recombination is modeled as follows:

$$R_{SRH} = \frac{pn - n_{ie}^2}{\mathbf{TAUP0}\left[n + n_{ie}\exp\left(\frac{\mathbf{ETRAP}}{kT_L}\right)\right] + \mathbf{TAUN0}\left[p + n_{ie}\exp\left(\frac{-\mathbf{ETRAP}}{kT_L}\right)\right]} \qquad 3\text{-}211$$

where ETRAP is the difference between the trap energy level and the intrinsic Fermi level, T_L is the lattice temperature in degrees Kelvin and TAUN0 and TAUP0 are the electron and hole lifetimes. This model is activated with the SRH parameter of the MODELS statement. The electron and hole lifetime parameters TAUN0 and TAUP0 are user definable on the MATERIAL statement. The default values for carrier lifetimes are shown in Table 3-35. Materials other than silicon will have different defaults and a full description of these are given in Appendix B.

Table 3-35. User Specifiable Parameters for Equation 3-211.

Statement	Parameter	Default	Units
MATERIAL	ETRAP	0	eV
MATERIAL	TAUN0	1e-7	s
MATERIAL	TAUP0	1e-7	s

Note: This model only presumes one trap level which by default is ETRAP=0 and corresponds to the most efficient recombination centre. If the TRAP statement is used to define specific trap physics then separate SRH statistics are implemented as described earlier in trap description earlier in this chapter.

SRH Concentration Dependent Lifetime Model

The constant carrier lifetimes that are used in the SRH recombination mode above may be made a function of impurity concentratrion after Roulston [11], using the equations;

$$R_{SRH} = \frac{pn - n_{ie}^2}{\tau_p\left[n + n_{ie}\exp\left(\frac{\mathbf{ETRAP}}{kT_L}\right)\right] + \tau_n\left[p + n_{ie}\exp\left(\frac{-\mathbf{ETRAP}}{kT_L}\right)\right]} \quad 3\text{-}212$$

$$\tau_n = \frac{\mathbf{TAUN0}}{1 + N/(\mathbf{NSRHN})} \quad 3\text{-}213$$

$$\tau_p = \frac{\mathbf{TAUP0}}{1 + N/(\mathbf{NSRHP})} \quad 3\text{-}214$$

where N is the local (total) impurity concentration. The parameters TAUN0, TAUP0, NSRHN and NSRHP may be user defined on the MATERIAL statement and have the default values shown in Table 3-36. This model is activated with the CONSRH parameter of the MODELS statement.

Table 3-36. User Specifiable Parameters for Equations 3-212 to 3-214.

Statement	Parameter	Default	Units
MATERIAL	TAUN0	1e-7	s
MATERIAL	NSRHN	5e16	cm^{-3}
MATERIAL	TAUP0	1e-7	s
MATERIAL	NSRHP	5e16	cm^{-3}

Klaassen's Concentration Dependent Lifetime Model

The Klaassen concentration and temperature dependent SRH lifetime model [115] is enabled by setting the KLASRH logical parameter on the MODELS statement. The lifetimes for electrons and holes in this model are given by the equations:

$$\mathbf{TAUN0}^{-1} = (\mathbf{KSRHTN}^{-1} + \mathbf{KSRHCN} \times N)\left(\frac{300}{T_L}\right)^{\mathbf{KSRHGN}} \quad 3\text{-}215$$

$$\mathbf{TAUP0}^{-1} = (\mathbf{KSRHTP}^{-1} + \mathbf{KSRHCP} \times N)\left(\frac{300}{T_L}\right)^{\mathbf{KSRHGP}} \quad 3\text{-}216$$

where N is the local (total) impurity concentration. The default values for the user definable parameters KSRHTN, KSRHTP, KSRHCN, KSRHCP, KSRHGN and KSRHGP are given in Table 3-37. These values may be modified on the MATERIAL statement.

Table 3-37. User Specifiable Parameters for Equations 3-215 to 3-216.

Statement	Parameter	Default	Units
MATERIAL	KSRHTN	2.5e-3	s
MATERIAL	KSRHTP	2.5e-3	s
MATERIAL	KSRHCN	3.0e-13	cm^3/s
MATERIAL	KSRHCP	11.76e-13	cm^3/s
MATERIAL	KSRHGN	1.77	
MATERIAL	KSRHGP	0.57	

Trap-Assisted Tunneling

In a strong electric field, electrons can tunnel through the bandgap via trap states. This trap-assisted tunneling mechanism, which is enables by specifying TRAP.TUNNEL on the MODELS statement, is accounted for by modifying the Schockley-Read-Hall recombination model.

$$R_{SRH} = \frac{pn - n_{ie}^2}{\frac{\mathbf{TAUP0}}{1+\Gamma_p}\left[n + n_{ie}\exp\left(\frac{\mathbf{ETRAP}}{kT_L}\right)\right] + \frac{\mathbf{TAUN0}}{1+\Gamma_n}\left[p + n_{ie}\exp\left(\frac{-\mathbf{ETRAP}}{kT_L}\right)\right]} \quad 3\text{-}217$$

where Γ_n is the field-effect enhancement functions for electrons and Γ_p is the corresponding function for holes. If the concentration dependent lifetime is used, the terms TAUN0 and TAUP0 are replaced by τ_n and τ_p as in equations 3-213 and 3-214. Two different enhancement functions are used, one corresponding to low values of electric field and the other to high field values [67].

The low field case corresponds to $K_{n,p} > \frac{2\Delta E_{n,p}}{3kT}$, where:

$$K_{n,p} = \frac{8\pi\sqrt{2\,\mathbf{MASS.TUNNEL}\,m_0\,\Delta E_{n,p}^3}}{3qh|E|} \quad 3\text{-}218$$

and q is the electronic charge, h is Planck's constant, m_0 is the rest mass of an electron and MASS.TUNNEL is the effective mass. The parameter MASS.TUNNEL may be set on the MODELS statement. The term $\Delta E_{n,p}$ is given by:

$$\Delta E_n = \begin{cases} E_C - E_{fn} & E_T \le E_{fn} \\ E_C - E_T & E_T > E_{fn} \end{cases} \quad 3\text{-}219$$

and

$$\Delta E_p = \begin{cases} E_{fp} - E_V & E_T > E_f \\ E_C - E_T & E_T \le E_f \end{cases} \quad 3\text{-}220$$

where E_C is the conduction band edge, E_V is the valence band edge, E_{fn} is the electron quasi-Fermi level, E_{fp} is the hole quasi-Fermi level and E_T is the energy of the trap level. The electron and hole field enhancement factors then become:

$$\Gamma_{n,p} = 2\sqrt{3\pi}\frac{|E|}{E_\Gamma}\exp\left[\left(\frac{E}{E_\Gamma}\right)^2\right] \quad 3\text{-}221$$

where

$$E_\Gamma = \frac{2\pi\sqrt{24\ \mathbf{MASS.TUNNEL}\ m_0(kT_L)^3}}{qh} \quad 3\text{-}222$$

and k is Boltzmann's constant and T_L is the lattice temperature and *E* is the electric field.

At high values of electric field where $K_{n,p} \le \frac{2\Delta E_{n,p}}{3kT_L}$, the enhancement terms becomes:

$$\Gamma_{n,p} = \frac{\Delta E_{n,p}}{2kT_L}\sqrt{\frac{\pi}{a}}\exp\left[\frac{b^2 - ac}{a}\right]\mathrm{erfc}\left(\frac{b}{\sqrt{a}}\right) \quad 3\text{-}223$$

where:

$$a = \frac{3}{8}K_{n,p},$$

$$b = \frac{\Delta E_{n,p}}{2kT_L} - \frac{3}{4}K_{n,p} \text{ and}$$

$$c = K_{n,p} - \frac{\Delta E_{n,p}}{kT_L}.$$

Table 3-38. User Specifiable Parameters for Equations 3-218 and 3-222.

Statement	Parameter	Default	Units
MATERIAL	MASS.TUNNEL	0.25	

Radiative (Direct) Recombination

Photon transitions occur when a direct transition from the valence band to the conduction band is possible. This process normally only occurs in narrow bandgap materials such as GaAs. The radiative, or direct, recombination model may be activated with the parameter OPTR on the MODELS statement. This model has the form:

$$R_{direct} = \mathbf{COPT}\,(pn - n_{ie}^2) \qquad 3\text{-}224$$

where the parameter COPT is user definable on the MATERIAL statement.

Note: In silicon direct band-to-band recombination is insignificant for almost every imaginable situation. This model should, therefore, only be used for narrow bandgap materials.

Auger Recombination

Auger recombination occurs through a three particle transition whereby a mobile carrier is either captured or emitted. The underlying physics for such processes is unclear and normally a more qualitative understanding is sufficient [3].

Standard Auger Model

Auger recombination is commonly modeled using the expression:

$$R_{Auger} = \mathbf{AUGN}\,(pn^2 - nn_{ie}^2) + \mathbf{AUGP}\,(np^2 - pn_{ie}^2) \qquad 3\text{-}225$$

where the model parameters AUGN and AUGP are user definable on the MATERIAL statement and have the silicon default parameters shown in Table 3-39. This model may be activated with the parameter AUGER of the MODELS statement.

Table 3-39. User Specifiable Parameters for Equation 3-225.

Statement	Parameter	Default	Units
MATERIAL	AUGN	8.3e-32	cm^6/s
MATERIAL	AUGP	1.8e-31	cm^6/s

Klaassen's Carrier Concentration Dependent Model

The form of the Klaassen Auger recombination model [116] is:

$$R_{Auger} = C_n(pn^2 - nn_{ie}^2) + C_p(np^2 - pn_{ie}^2) \qquad 3\text{-}226$$

where the Auger coefficients are concentration dependent according to:

$$C_n = p^2 \, \mathbf{KAUGCN}\left(\frac{T_L}{300}\right)^{\mathbf{KAUGDN}} \quad 3\text{-}227$$

$$C_p = n^2 \, \mathbf{KAUGCP}\left(\frac{T_L}{300}\right)^{\mathbf{KAUGDP}} \quad 3\text{-}228$$

where the model parameters KAUGCN, KAUGCP, KAUGDN and KAUGDP are user definable on the MATERIAL statement and have the defaults shown in Table 3-40. This model is activated by specifying the KLAAUG paramater of the MODELS statement.

Table 3-40. User Specifiable Parameters for Equation 3-227 and 3-228.

Statement	Parameter	Default	Units
MATERIAL	KAUGCN	1.83e-31	cm^6/s
MATERIAL	KAUGCP	2.78e-31	cm^6/s
MATERIAL	KAUGDN	1.18	
MATERIAL	KAUGDP	0.72	

Narrow Bandgap Auger Model

An alternative model for the Auger recombination coefficients that is more suitable for modelling Auger processes in narrow bandgap semiconductors can be enabled by setting the parameters KAGUN and KAGUP on the MODELS statement. The model in ATLAS is a simplification of that by Beattie [146] and takes the form:

$$R_{Auger} = C_n(pn^2 - nn_{ie}^2) + C_p(np^2 - pn_{ie}^2) \quad 3\text{-}229$$

where the Auger coefficients are concentration dependent according to:

$$C_n = \frac{\mathbf{AUGN}}{1 + \mathbf{AUGKN}\ n} \quad 3\text{-}230$$

$$C_p = \frac{\mathbf{AUGP}}{1 + \mathbf{AUGKP}\ p} \quad 3\text{-}231$$

where n and p are the electron and hole carrier concentrations and the new parameters AUGKN and AUGKP are user definable on the MODELS statement.

Surface Recombination

In addition to generation-recombination within the bulk of the semiconductor, electrons or holes may recombine or be generated at interfaces. The rate of surface recombination may be even greater than within the bulk. The standard method is to model interface recombination in a similiar manner as the bulk gerneration-recombination rate:

$$R_{surf} = \frac{pn - n_{ie}^2}{\tau_p^{eff}\left[n + n_{ie}\exp\left(\frac{\mathbf{ETRAP}}{kT_L}\right)\right] + \tau_n^{eff}\left[p + n_{ie}\exp\left(\frac{-\mathbf{ETRAP}}{kT_L}\right)\right]} \quad 3\text{-}232$$

$$\frac{1}{\tau_n^{eff}} = \frac{1}{\tau_n^i} + \frac{d_i}{A_i}\mathbf{S.N} \quad 3\text{-}233$$

$$\frac{1}{\tau_p^{eff}} = \frac{1}{\tau_p^i} + \frac{d_i}{A_i}\mathbf{S.P} \quad 3\text{-}234$$

where τ^i_n is the bulk lifetime calculated at node i along the interface and which may be a function of the impurity concentration as well. The parameters d_i and A_i are the length and area of the interface for node i. The two parameters S.N and S.P are the recombination velocities for electrons and holes respectively, which are user definable on the INTERFACE statement. The parameters X.MIN, X.MAX, Y.MIN and Y.MAX can also be set on the INTERFACE statement to define the region in which the specified values of the surface recombination velocities apply. This model is activated by the presence of the recombination velocities on the INTERFACE statement.

Table 3-41. User Specifiable Parameters for Equations 3-232 to 3-234.

Statement	Parameter	Default	Units
INTERFACE	S.N	0	cm/s
INTERFACE	S.P	0	cm/s

Impact Ionization Models

Overview

In any space charge region with a sufficiently high revere bias, the electric field will be high enough to accelerate free carriers up to a point where they will have acquired sufficient energy to generate more free carriers when in collision with the atoms of the crystal. In order to acquire sufficient energy two principle conditions must be met:

- the electric field must be sufficiently high
- the distance between the collisions of the free carrier must be enough to allow acceleration to a sufficiently high velocity

In other words the carrier must gain the ionisation energy E_i between collisions. If the generation rate of these free carriers is sufficiently high this process may eventually lead to avalanche breakdown.

The general impact ionisation process can be described by the equation;

$$G = \alpha_n J_n + \alpha_p J_p \quad 3\text{-}235$$

where G is the total generation rate of electron-hole pairs, $\alpha_{n,p}$ are the ionisation coefficient for electrons and holes and $J_{n,p}$ are their current densities. The ionisation coefficient represents the

number of electron-hole pairs generated by a carrier per unit distance travelled. The accurate calculation of this parameter has been the subject of much research work as it is vital if the effects related to impact ionization, such as substrate current and device breakdown, are to be simulated. These models may be classified into two main types: local and nonlocal models. The former assume that ionization at any particular point within the device is a function only of the electric field at that position. Nonlocal models on the other hand, perform a more rigorous approach by taking into account the energy that the carrier gains.

Local Electric Field Models

Selberherr's Impact Ionization Model

The ionization rate model proposed by Selberherr [3] is a variation of the classical Chynoweth model [127]. It is activated by the SELB parameter of the IMPACT statement and is based upon the following expressions:

$$\alpha_n = \mathbf{AN}\exp\left[-\left(\frac{\mathbf{BN}}{E}\right)^{\mathbf{BETAN}}\right] \quad 3\text{-}236$$

$$\alpha_p = \mathbf{AP}\exp\left[-\left(\frac{\mathbf{BP}}{E}\right)^{\mathbf{BETAP}}\right] \quad 3\text{-}237$$

where E is the electric field in the direction of current flow at a particular position in the structure and the parameters AN, AP, BN, BP, BETAN and BETAP may be defined on the IMPACT statement and have the default values shown in Table 3-42. In the case of AN, AP, BN and BP it is also possible to define a value of electric field EGRAN V/cm where for electric fields <EGRAN V/cm the parameters are AN1, AP1, BN1, BP1 whilst for electric fields >EGRAN V/cm the parameters become AN2, AP2, BN2 and BP2.

The AN and BN parameters are also a function of the lattice temperature in this model. The temperature dependence of these coefficients is defined as follows;

$$\mathbf{AN} = \mathbf{AN_{1,2}}\left(1 + \mathbf{A.NT}\left[\left(\frac{T_L}{300}\right)^{\mathbf{M.ANT}} - 1\right]\right) \quad 3\text{-}238$$

$$\mathbf{AP} = \mathbf{AP_{1,2}}\left(1 + \mathbf{A.PT}\left[\left(\frac{T_L}{300}\right)^{\mathbf{M.APT}} - 1\right]\right) \quad 3\text{-}239$$

$$\mathbf{BN} = \mathbf{BN_{1,2}}\left(1 + \mathbf{B.NT}\left[\left(\frac{T_L}{300}\right)^{\mathbf{M.BNT}} - 1\right]\right) \quad 3\text{-}240$$

$$\mathbf{BP} = \mathbf{BP_{1,2}}\left(1 + \mathbf{B.PT}\left[\left(\frac{T_L}{300}\right)^{\mathbf{M.BPT}} - 1\right]\right) \quad 3\text{-}241$$

The parameters associated with these equations are shown in Table 3-43.

Table 3-42. User Definable Parameters in the Selberherr Impact Ionization Model.		
Statement	**Parameter**	**Default**
IMPACT	AN1	$7.03*10^5$ cm^{-1}
IMPACT	AN2	$7.03*10^5$ cm^{-1}
IMPACT	AP1	$6.71*10^5$ cm^{-1}
IMPACT	AP2	$6.71*10^5$ cm^{-1}
IMPACT	BN1	$1.131*10^6$V/cm
IMPACT	BN2	$1.231*10^6$ V/cm
IMPACT	BP1	1.693*106 V/cm
IMPACT	BP2	2.036*106 V/cm
IMPACT	BETAN	1.0
IMPACT	BETAP	1.0
IMPACT	EGRAN	4*105 V/cm

Table 3-43. Temperature Coefficient Parameters of the Selberherr Impact Ionization Model for Silicon in Equations 3-238 to 3-241.		
Statement	**Parameter**	**Default**
IMPACT	A.NT	0.588
IMPACT	B.NT	0.248
IMPACT	A.PT	0.588
IMPACT	B.PT	0.248
IMPACT	M.ANT	1.0
IMPACT	M.BNT	1.0
IMPACT	M.APT	1.0
IMPACT	M.BPT	1.0

Grant's Impact Ionization Model

The second ionization model has the same form as the Selberherr model but a simpler implementation:

$$\alpha_n = \mathbf{AN}\exp\left[-\left(\frac{\mathbf{BN}}{E}\right)\right] \quad 3\text{-}242$$

$$\alpha_p = \mathbf{AP}\exp\left[-\left(\frac{\mathbf{BP}}{E}\right)\right] \quad 3\text{-}243$$

This implementaion has three key differences:

- the model has a low field, an intermediate field and a high field region
- the coefficients for silicon are different
- there is no temperature dependence

This model was developed after investigations by Baraff [128] suggested the existence of a low, intermediate and high field response region for electron and hole ionisation rates. The coefficients implemented into this model match the experimental data of Grant [2] which suggested that the three different regions existed.

This model is activated with the parameter GRANT on the MODEL statement. The model parameters AN, AP, BN and BP are **NOT** user definable. Instead, the three electric field regions have in-built values as follows:

1) **Low Electric Field** E < 2.4e5 V/cm 3-244

AN = 2.6e6 AP = 2.0e6

BN = 1.43e6 BN = 1.97e6

2) **Intermediate Electric Field** 2.4e5 > E > 5.3e5 V/cm 3-245

AN = 6.2e5 AP= 2.0e6

BN = 1.08e6 BP = 1.97e6

3) **High Electric Field** E > 5.3e5 V/cm 3-246

AN = 5.0e5 AP = 5.6e5

BN = 9.9e5 BP = 1.32e6

Crowell-Sze Impact Ionization Model

Crowell and Sze [1] have proposed a more physical, but complicated, relationship between the electric field and the ionization rates. This model represents ionization coefficients as follows:

$$\alpha_{n,p} = \frac{1}{\lambda_{n,p}}\exp[C_o(r) + C_1(r)x + C_2(r)x^2] \quad 3\text{-}247$$

$$C_0 = -1.92 + 75.5r - 75.7r^2 \quad 3\text{-}248$$

$$C_1(r) = -1.75\times10^{-2} - 11.9r + 46r^2 \quad 3\text{-}249$$

$$C_2(r) = 3.9\times10^{-4} - 1.17r + 11.5r^2 \quad 3\text{-}250$$

where:

$$r = \frac{E_r}{E_i}; \quad x = \frac{E_i}{q\lambda_{n,p}E} \tag{3-251}$$

$$E_r = 0.063 \text{ eV} \tag{3-252}$$

$$E_i = \begin{cases} 1.1\text{eV} & \text{for electrons} \\ 1.8\text{eV} & \text{for holes} \end{cases} \tag{3-253}$$

$$\lambda_n^o = \mathbf{LAMDAE}\frac{\tanh[qE_r/2kT_L]}{\tanh[qE_r/2k300]} \tag{3-254}$$

$$\lambda_n^o = \mathbf{LAMDAH}\frac{\tanh[qE_r/2kT_L]}{\tanh[qE_r/2k300]} \tag{3-255}$$

The Crowell-Sze model for impact ionization is selected by setting the CROWELL parameter of the IMPACT statement.

Table 3-44. Crowell-Sze Impact Ionization Model Parameters.

Statement	Parameter	Default
IMPACT	LAMDAE	6.2*10-7cm
IMPACT	LAMDAH	3.8*10-7 cm

Non Local Energy Dependent Models

All local electric field based models will normally overestimate the rate of impact ionization. This occurs because lucky electron theory inherently assumes that a carrier is traveling through a constant electric field E. As a result it will predict a distance $\Delta x = E_i/qE$ over which the carrier will gain the ionization energy E_i. However, in real devices the electric field is never constant but is normally sharply peaked at metallurgical junctions. Therefore, as a carrier passes through the peaked electric field the lucky electron model will predict the ionization distance Δx to be too small. As a result the ionization rate is overestimated. The effect of this is that all the simulated breakdown voltages will be underestimated and substrate currents overestimated.

The energy balance model can be used to improve the simulation of impact ionization by implementing ionization models based upon the carrier temperature instead of the electric field. The carrier temperature is a more meaningful basis as the velocity-field relationship is more closely modeled. This allows a "nonlocal" dependence on the electric field within the impact ionization model. Energy balance models will therefore result in more accurate simulations of breakdown voltage and substrate current. Two different impact ionization models have been implemented into ATLAS, the first is based upon the classical Chynoweth relationship, modified to include carrier temperature, but the second is a more advanced non-Maxwellian approach based upon carrier temperatures.

When the energy balance transport model is applied only two impact ionization models are available; the Toyabe model and the Concannon model.

Toyabe Impact Ionization Model

The temperature dependent impact ionization model is founded on the Selberherr model and is similar to that suggested by Toyabe [5]. The carrier temperature is used to calculate an effective electric field based upon the homogeneous temperature-field relation. To maintain self-consistency within the energy balance model this is the same relationship used for the effective electric field within the carrier temperature dependent mobility. This model is the default model for impact ionization and is activated with the TOYABE or SELB parameters on the IMPACT statement. The ionization rates now have the form:

$$\alpha_n = \mathbf{AN}\exp\left(\frac{-\mathbf{BN}}{E_{eff,\,n}}\right) \quad \text{3-256}$$

$$\alpha_p = \mathbf{AP}\exp\left(\frac{-\mathbf{BP}}{E_{eff,\,p}}\right) \quad \text{3-257}$$

where the model parameters AN, AP, BN and BP are user definable on the IMPACT statement and have the default values shown in Table 3-42. The effective electric field is calculated according to:

$$E_{eff,\,n} = \frac{3}{2}\frac{kT_n}{q\mathbf{LREL.EL}} \quad \text{3-258}$$

$$E_{eff,\,p} = \frac{3}{2}\frac{kT_p}{q\mathbf{LREL.HO}} \quad \text{3-259}$$

where the energy relaxation lengths LREL.EL and LREL.HO may be explicitly defined on the IMPACT statement or may be calculated according to:

$$\mathbf{LREL.EL} = \mathbf{VSATN} \times \mathbf{TAUSN} \quad \text{3-260}$$

$$\mathbf{LREL.HO} = \mathbf{VSATP} \times \mathbf{TAUSP} \quad \text{3-261}$$

where VSATN and VSATP are the saturation velocities for electrons and holes.

Table 3-45. User Specifiable Parameters for Equations 3-258 - 3-261.

Statement	Parameter	Units
IMPACT	LREL.EL	μm
IMPACT	LREL.HO	μm
MATERIAL	VSAT	cm/s
MATERIAL	VSATN	cm/s

Table 3-45. User Specifiable Parameters for Equations 3-258 - 3-261.		
Statement	**Parameter**	**Units**
MATERIAL	VSATP	cm/s
IMPACT	TAUSN	s
IMPACT	TAUSP	s

Note: As an added level of flexibility, the relaxation times used for the energy balance equation and those used in the impact ionization model have been separated into two user-definable parameters. In contrast to TAUREL.EL and TAUREL.HO which are used in different formulae, the parameters TAUSN and TAUSP are only applicable in the impact ionization expression 3-260 and 3-261. By default, TAUREL.EL=TAUSN and TAUREL.HO=TAUSP.

It can also be argued that the parameters AN, AP, BN, and BP should also be a function of the carrier temperature. However, no clear theoretical basis for this has been proposed and accepted. Instead the C-Interpreter within ATLAS has been extended to include two C-interpreter functions. These functions are specified via the F.EDIIN and F.EDIIP parameters of the IMPACT statement. These parameters specify the filename of a text file containing a C-interpreter function that describes the dependence of the model parameters AN, AP, BN and BP as a function of the carrier temperatures. These values will then be used within Toyabe's energy dependent impact ionization model.

Concannon's Impact Ionization Model

The previous nonlocal impact ionisation model inherently assumes a Maxwellian shape to the distribution of hot carriers. Recent work by Fiegna [129] using Monte Carlo simulations suggests a non-Maxwellian high energy tail to the energy distribution function. To more accurately model these effects a non-Maxwellian based model from Concannon [112] has been implemented. Based upon this energy distribution the model calculates the probability of a carrier having sufficient energy to cause impact ionisation. The model results show good agreement with measured results for a 0.9μm flash EPROM device [120].

The Concannon substrate current is enabled for the electron and hole continuity equations by specifying the N.CONCANNON and P.CONCANNON parameters of the IMPACT statement.

The generation rate is a function of the carrier temperature and concentration, and is given by:

$$G_n(x,y) = \mathbf{CSUB.N} \times n \int_{\mathbf{ETH.N}}^{\infty} F(\varepsilon, T_n(x,y)))d\varepsilon \qquad 3\text{-}262$$

$$G_p(x,y) = \mathbf{CSUB.P} \times p \int_{\mathbf{ETH.P}}^{\infty} F(\varepsilon, T_p(x,y)))d\varepsilon \qquad 3\text{-}263$$

where n(x,y) and p(x,y) are the electron and hole carrier concentrations within the semiconductor, ε is energy, $T_n(x,y)$ and $T_p(x,y)$ are the electron and hole carrier temperatures in the semiconductor, CSUB.N, CSUB.P, ETH.N, and ETH.P are user specifiable parameters as given in Table 3-46.

Table 3-46. User Specifiable Parameters for Equations 3-262 and 3-263			
Statement	**Parameter**	**Default**	**Units**
IMPACT	CSUB.N	2.0e14	
IMPACT	CSUB.P	4.0e14	
IMPACT	ETH.N	1.8	eV
IMPACT	ETH.P	3.5	eV

The function F(ε,T) in Equations 3-262 and 3-263 is given by the product of the density of states function, g(ε), and the energy distribution function f(ε) as:

$$F = \frac{g(\varepsilon)f(\varepsilon)}{\int_0^\infty g(\varepsilon)f(\varepsilon)} \quad 3\text{-}264$$

The density of states function is given by:

$$g(\varepsilon) = \varepsilon^{1.25} \quad 3\text{-}265$$

The energy distribution functions for electrons and holes are:

$$f_n(\varepsilon) = \left[\exp\left(\frac{-\mathbf{CHIA}\ \varepsilon^3}{T_n^{1.5}}\right) + \mathbf{C0}\exp\left(\frac{-\mathbf{CHIB}\ \varepsilon^3}{T_n^{1.5}}\right)\right] \quad 3\text{-}266$$

$$f_p(\varepsilon) = \exp\left(\frac{-\mathbf{CHI.HOLES}\ \varepsilon^3}{T_p^{1.5}}\right) \quad 3\text{-}267$$

where ε is energy, $T_{n,p}$ are the carrier temperatures, CHIA, CHIB, and C0 are user specifiable parameters as given in Table 3-47.

Table 3-47. User Definable Parameters for the Energy Distribution Functions			
Statement	**Parameter**	**Default**	**Units**
IMPACT	CHIA	3.0e5	
IMPACT	CHIB	5.0e4	
IMPACT	C0	2.5e-10	
IMPACT	CHI.HOLES	4.6e4	

Two other parameters of the IMPACT statement are user definable that may effect the result of the numeric integration. The ENERGY.STEP parameter specifies the energy step size in eV used during the numeric integration. The default step size is 25 meV. The INFINITY parameter sets the upper limit of

the integration and specifies ratio of the increment added to the integral divided by the current value of the integral. The default value of the INFINITY parameter is 0.001.

Note: To maintain self-consistent results it is important that this model is implemented if the Concannon model is being used for the simulation of gate current.

Band-to-Band Tunneling

If a sufficiently high electric field exists within a device local band bending may be sufficient to allow electrons to tunnel, by internal field emission, from the valence band into the conduction band. An additional electron is therefore generated in the conduction band and a hole in the valence band. This generation mechanism is implemented into the right-hand side of the continuity equations. The tunneling generation rate is [69,70] as:

$$G_{BBT} = \mathbf{BB.A}\ E^{\mathbf{BB.GAMMA}} \exp\left(-\frac{\mathbf{BB.B}}{E}\right) \quad 3\text{-}268$$

where E is the magnitude of an electric field and BB.A, BB.B, and BB.GAMMA are user definable parameters. In ATLAS there are three different sets of values that may be applied to the model parameters.

The model parmaeters can be set to the standard mode [69] by specifying BBT.STD on the MODELS statement. The parameter defaults for the standard model are as follows:

BB.A = 9.66e18 V/cm BB.B= 3.0e7 BB.GAMMA= 2.0

The model parameters may also be set to the Klaassen model [70] by specifying BBT.KL on the MODELS statement. The parameter defaults for the Klaassen model are as follows:

BB.A = 4.00e14 V/cm BB.B = 1.97e7 BB.GAMMA= 2.5

The third alternative allows these model parameters to be calculated from first principles by specifying the parameter AUTOBBT on the MODELS statement. In this case the parameters are calculated according to:

$$\mathbf{BB.A} = \frac{q^2\sqrt{(2 \times \mathbf{MASS.TUNNEL}\ m_0)}}{h^2\sqrt{E_g}} \quad 3\text{-}269$$

$$\mathbf{BB.B} = \frac{\pi^2\ \mathbf{EG300}^{\frac{3}{2}}\sqrt{\frac{\mathbf{MASS.TUNNEL}\ m_0}{2}}}{qh} \quad 3\text{-}270$$

$$\mathbf{BB.GAMMA} = 2 \quad 3\text{-}271$$

where q is the electronic charge, h is Planck's constant, E_g is the energy bandgap, m_0 is the rest mass of an electron and MASS.TUNNEL is the effective mass. The parameter MASS.TUNNEL may be set on the MODELS statement and the bandgap at 300K, EG300, is defined on the MATERIAL statement.

Table 3-48. User Definable Parameters in the Band-toBand Tunneling Model			
Statement	**Parameter**	**Default**	**Units**
MODEL	BB.A	9.66e18	$V^{-2}s^{-1}cm^{-1}$
MODEL	BB.B	3e17	V/cm
MODEL	BB.GAMMA	2.0	

Gate Current Models

Overview

In devices that have a metal-insulator-semiconductor (MIS) formation, the conductance of the insulating film would ideally be considered as zero. However, for the sub 0.5um generation of MOS devices there is now considerable conductance being measured on the gate contacts. This gate current has resulted in two major consequences, one negative and one positive.

On the negative side, the gate current is responsible for the degradation in device operating characteristics with time. This “reliability” issue is of considerable importance as the lifetime of electronic parts has to be guaranteed. Reliability may be simulated within the Silvaco suite of tools for device level reliability which are described in a later chapter.

On the positive side, the existence of this gate current has caused the proliferation of the non-volatile memory market. These devices use the existence of gate current to program and erase the charge on a “floating” contact. This concept has resulted in a variety of different devices such as FLASH, FLOTOX, EEPROM, etc. All such devices rely on the physics of the gate current process for their existence.

There are a variety of different conduction mechanisms within an insulating layer [13], but for the case of nonvolatile memory, only two mechanism are relevent: Fowler-Nordheim tunneling and hot carrier injection. Models for these two injection processes are described in the following sections. In the case of hot electron injection two models are available the lucky electron model and the Concannon gate current model.

Fowler-Nordheim Tunneling

If the electric field across an insulator is sufficiently high then it may cause tunneling of electrons from the semiconductor (or metal) Fermi level into the insulator conduction band. This process is strongly dependent on the applied electric field but is independent of the ambient temperature.

The Fowler-Nordheim equation [69] expresses tunnel current density through the oxide as:

$$J_{FN} = \mathbf{F.AE}\ E^2 \exp\left(-\frac{\mathbf{F.BE}}{E}\right) \quad 3\text{-}272$$

$$J_{FP} = \mathbf{F.AH}\ E^2 \exp\left(-\frac{\mathbf{F.BH}}{E}\right) \quad 3\text{-}273$$

where E specifies the magnitude of the electric field in the oxide. The model parameters F.AE, F.AH, F.BE, and F.BH may be defined on the MODELS statement but have the default values obtained from Keeney, Piccini, and Morelli [69] shown in Table 3-49.

Table 3-49. User Specifiable Parameters for Equation			
Symbol	**Statement**	**Parameter**	**Default Values**
A_{FN}	MODELS	F.AE	$1.82 \cdot 10^{-7}$
B_{FN}	MODELS	F.BE	$1.90 \cdot 10^{8}$
A_{FH}	MODELS	F.AH	$1.82 \cdot 10^{-7}$
B_{FH}	MODELS	F.BH	$1.90 \cdot 10^{8}$

The Fowler-Nordheim model in ATLAS has been implemented as either a post processing function or as a self-consistent scheme with the other equations. The post processing option may be chosen by specifying the paramter FNPP on the MODEL statement. The self consistent scheme is activated by specifying the parameter FNORD on the MODEL statement.

For either model the implementation scheme is the same. Each electrode-insulator and insulator-semiconductor interface is divided into discrete segments which are based upon the mesh. For each insulator-semiconductor segment the Fowler-Nordheim current is calculated as described above. This current will then be added to a segment on the electrode-insulator boundary. Two schemes have been implemented to find out to which segment this current should be added.

The default model that calculates which electrode segment receives the Fowler-Nordheim current follows the path of the electric field vector at the semiconductor-insulator interface. The first electrode-insulator segment that is found along this trajectory, provided no other semiconductors or metals are found along the trajectory, will receive the Fowler-Nordheim current.

A second model may be chosen using the NEARFLG parameter of the MODEL statement. In this case the electrode-insulator segment found closest to the semiconductor-insulator segment will receive the Fowler-Nordheim current.

The total current on the gate electrode is then the sum of the currents from all the individual segements around the electrode boundary.

Note: Since Fowler-Nordheim tunneling current is responsible for EPROM and EEPROM cell erasure, this model should always be specified when performing erasure simulation. It is also recommended that the band to band tunneling model is included if Fowler-Norheim tunneling is being modelled.

Note: When simulating EPROM erasure in a transient analysis with this model, the floating contact charge becomes a function of the gate current. In this case the total current flowing into the floating electrode is multiplied by the time step to calculate the charge added to the electrode during that time step. The new value of charge is then used as the boundary condition for the next time step.

Lucky Electron Hot Carrier Injection Model

In the lucky-electron model it is proposed that an electron is emitted into the oxide by first gaining enough energy from the electric field in the channel to surmount the insulator/semiconductor barrier. Once the required energy to surmount the barrier has been obtained the electrons are redirected towards the insulator/semiconductor interface by some form of phonon scattering. When these

conditions are met then the carrier travelling towards the interface will have an additional probability that it will not suffer any additional collision through which energy could be lost.

The model implemented into ATLAS is a modified version of the model proposed by Tam [30] and is activated by the parameter HEI and HHI, for electron and hole injection respectively, on the MODELS statement. The gate electrode-insulator interface is subdivided into a number of discrete segments which are defined by the mesh. For each segment the lucky electron model is used to calculate the injected current into that segment. The total gate current is then the sum of all of the discrete values.

If we consider a discrete point on the gate electrode-insulator boundary we can write a mathematical formula for the current injected from the semiconductor. The formula calculates the injected gate current contribution from every node point within the semiconductor according to:

$$I_{inj} = \iint P_n(x, y)\left|\vec{J}_n(x, y)\right| dx\ dy + \iint P_p(x, y)\left|\vec{J}_p(x, y)\right| dx\ dy \qquad 3\text{-}274$$

where $J_{n,p}$ (x,y) are the electron and hole current densities at a point (x,y) within the semiconductor, and $P_{n,p}$ (x,y) are the probabilities that a fraction of this current reaches the gate oxide and is injected across into the gate electrode. The total probability $P_{n,p}$ (x,y) is defined by;

$$P_n(x, y) = P_{\phi B, n} P_{1, n} P_{2, n} / \mathbf{EG.ELINR} \qquad 3\text{-}275$$

$$P_p(x, y) = P_{\phi B, p} P_{1, p} P_{2, p} / \mathbf{IG.HLINR} \qquad 3\text{-}276$$

where E is the electric field parallel to the current flow, IG.ELINR and IG.HLINR are the electron and hole mean free path lengths between redirecting collisions. The three probability factors will now be described.

The probability $P\phi_B$ is the probability of a carrier gaining the energy ϕ_B by moving in, and parallel to, an electric field E, without suffering energy loss by optical phonon scattering and is given by:

$$P_{\phi_{B, n}} = 0.25\left(\frac{E\ \mathbf{IG.ELINF}}{\phi_{B, n}}\right)\exp\left(-\frac{\phi_{B, n}}{E\ \mathbf{IG.ELINF}}\right) \qquad 3\text{-}277$$

$$P_{\phi_{B, p}} = 0.25\left(\frac{E\ \mathbf{IG.HLINF}}{\phi_{B, p}}\right)\exp\left(-\frac{\phi_{B, p}}{E\ \mathbf{IG.HLINF}}\right) \qquad 3\text{-}278$$

where IG.ELINF and IG.HLINF are the mean free path lengths of electrons and holes for scattering by optical phonons. The barrier heights $\phi_{Bn,p}$ are defined according to:

$$\phi_{B, n} = \mathbf{IG.EB0} - \mathbf{IG.EBETA}\sqrt{E_\perp} - \mathbf{IG.EETA}E_\perp^{2/3} - \Delta\psi(x, y) \qquad 3\text{-}279$$

$$\phi_{B, p} = \mathbf{IG.HB0} - \mathbf{IG.HBETA}\sqrt{E_\perp} - \mathbf{IG.HETA}E_\perp^{2/3} - \Delta\psi(x, y) \qquad 3\text{-}280$$

where $E_\perp$ is the electric field perpendicular to the semiconductor-insulator interface. The traditional barrier heights, IG.EB0 and IG.HB0 are reduced to take account of three effects. The first is due to Schottky barrier lowering which depends on the perpendicular electric field at the semiconductor-insulator interface. The second takes account of tunneling through the gate oxide by reducing the barrier height. The third takes into account that a potential difference exists between the semiconductor-insulator interface and the starting position of the hot carrier. By default this last effect

is disabled but may be enabled with the parameters E.BENDING and H.BENDING for electrons and holes respectively.

The second probability P_1 is the probability that no energy is lost by optical phonon scattering as the hot carrier travels towards the semiconductor-insulator interface after being redirected, and is given by:

$$P_{1,n} \sim \exp\left(-\frac{r}{\mathbf{IG.ELINF}}\right) \quad \text{3-281}$$

$$P_{1,p} \sim \exp\left(-\frac{r}{\mathbf{IG.HLINF}}\right) \quad \text{3-282}$$

where r is the distance from point of redirection to the semiconductor-insulator interface.

The final probability P_2 accounts for the probability of scattering in the image force potential well in the gate oxide and is given by;

$$P_{2,n} = \exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{\mathbf{PATH.N}}\right) \quad \text{for } \theta > \mathbf{THETA.N} \quad \text{3-283}$$

$$P_{2,n} = 0 \quad \text{for } \theta > \mathbf{THETA.N} \quad \text{3-284}$$

$$P_{2,p} = \exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{\mathbf{PATH.P}}\right) \quad \text{for } \theta > \mathbf{THETA.P} \quad \text{3-285}$$

$$P_{2,p} = 0 \quad \text{for } \theta < \mathbf{THETA.P} \quad \text{3-286}$$

where PATH.N and PATH.P are the electon and hole mean free path lengths within the oxide, ε_{ox} is the oxide permittivity and E_{ox} is the electric field in the oxide. The angle θ introduces an angle dependence which is based upon the work of Wada [131]. His experiments indicate a critical rejection angle, THETA.N and THETA.P, between the angle θ formed between the semiconductor-insulator interface and the electric field in the oxide. If the angle θ is less than the rejection angle then the electrons are repelled back to the substrate.

Table 3-50 lists the user definable model parameters which can be set on the MODEL statement, their default values and their units.

Table 3-50. User Definable Parameters in Concannon's Gate Current Model.			
Statement	**Parameter**	**Default**	**Units**
MODELS	IG.ELINR	$6.16x10^{-6}$	cm
MODELS	IG.HLINR	$6.16x10^{-6}$	cm
MODELS	IG.ELINF	$9.2x10^{-7}$	cm
MODELS	IG.HLINF	$9.2x10^{-7}$	cm
MODELS	IG.EBETA	$2.59x10^{-4}$	$(Vcm)^{1/2}$
MODELS	IG.HBETA	$2.59x10^{-4}$	$(Vcm)^{1/2}$
MODELS	IG.EETA	$2x10^{-5}$	$V^{1/3}cm^{2/3}$
MODELS	IG.HETA	$2x10^{-5}$	$V^{1/3}cm^{2/3}$
MODELS	PATH.N	$3.4x10^{-7}$	cm
MODELS	PATH.P	$2.38x10^{-7}$	cm
MODELS	THETA.N	60	degrees
MODELS	THETA.P	60	degrees
MODELS	IG.EB0	3.2	eV
MODELS	IG.HB0	4.0	eV

The implementation of this model is similiar to that for Fowler-Nordheim tunneling. Each electrode-insulator and insulator-semiconductor interface is divided into discrete segments which are based upon the mesh. For each insulator-semiconductor segment the Fowler-Nordheim current is calculated as described above. This current will then be added to a segment on the electrode-insulator boundary. Two schemes have been implemented to find out to which segment this current should be added.

The default model that calculates which electrode segment receives the hot carrier injected current follows the path of the electric field vector at the semiconductor-insulator interface. The first electrode-insulator segment that is found along this trajectory, provided no other semiconductors or metals are found along the trajectory, will receive the current.

A second model may be chosen using the NEARFLG parameter of the MODELS statement. In this case the electrode-insulator segment found closest to the semiconductor-insulator segment will receive the hot carrier injected current.

The total current on the gate electrode is then the sum of the currents from all the individual segements around the electrode boundary.

Note: When simulating EPROM programming with this model the floating contact charging is simulated in the transient mode. In this case the total current flowing into the floating electrode is multiplied by the time step to calculate the charge added to the electrode during that time step. The new value of charge is then used as the boundary condition for the next time step.

Concannon's Injection Model

The implicit assumption in the lucky electron approach is a Maxwellian shape for the energy distribution of the hot carriers. Recent work by Fiegna [129] using Monte Carlo simulations suggests a non-Maxwellian high energy tail to the distribution function. To more accurately model these effects a non-Maxwellian based model from Concannon [112] has been implemented. This model requires the solution to the energy balance equation for the carrier temperatures but has been implemented in a similaiar manner to the lucky electron model. The Concannon gate injection model may be specified with the parameters N.CONCANNON and P.CONCANNON on the MODELS statement. This choice of parameters will automatically activate the energy balance model.

The Concannon injection model has a similiar form to the lucky electron model. The injected current is calculated according to:

$$I_{inj} = \iint P_n(x, y)\ n(x, y)dx\ dy + \iint P_p(x, y)\ p(x, y)dx\ dy \qquad 3\text{-}287$$

where n(x,y) and p(x,y) are the carrier concentrations within the semiconductor. The probability functions $P_n(x,y)$ and $P_p(x,y)$ are now defined by:

$$P_n(x, y) = -q\ \mathbf{CGATE.N}\ P_{\phi_{B,n}} P_{1,n} P_{2,n} \qquad 3\text{-}288$$

$$P_p(x, y) = q\ \mathbf{CGATE.P}\ P_{\phi_{B,p}} P_{1,p} P_{2,p} \qquad 3\text{-}289$$

where q is the electronic charge and the parameters CGATE.N and CGATE.P are user definable on the MODEL statement. The three probability functions, in equations 3-288 and 3-289, shall now be described.

The probability that a carrier has sufficient energy to surmount the insulator-semiconductor barrier of height ϕ_B is now defined as a function of energy. The probability now has the form:

$$P_{\phi_{B,n}} = \int_{\phi_{B,n}}^{\infty} v_{\perp}(\varepsilon) F(\varepsilon, T_n(x, y))d \qquad 3\text{-}290$$

$$P_{\phi_{B,p}} = \int_{\phi_{B,p}}^{\infty} v_{\perp}(\varepsilon) F(\varepsilon, T_p(x, y))d \qquad 3\text{-}291$$

where $v_{\perp}(\varepsilon)$ is the perpendicular velocity of a hot carrier and defines the probability of a hot carrier with an energy ε travelling in the direction of the insulator-semiconductor. The barrier heights $\phi_{Bn,p}$ are defined according to:

$$\phi_{B,n} = \mathbf{PEFF.N} - \mathbf{IG.EBETA}\sqrt{E_{\perp}} - \mathbf{IG.EETA}\ E_{\perp}^{2/3} - \Delta\psi(x, y) \qquad 3\text{-}292$$

$$\phi_{B,p} = \mathbf{PEFF.P} - \mathbf{IG.HBETA}\sqrt{E_{\perp}} - \mathbf{IG.HETA}\ E_{\perp}^{2/3} - \Delta\psi(x, y) \qquad 3\text{-}293$$

where $E_{\perp}$ is the electric field perpendicular to the semiconductor-insulator interface. The traditional barrier heights, IG.EB0 and IG.HB0 are reduced to take account of three effects. The first is due to Schottky barrier lowering which depends on the perpendicular electric field at the semiconductor-insulator interface. The second takes account of tunneling through the gate oxide by reducing the barrier height. The third takes into account that a potential difference exists between the semiconductor-insulator interface and the starting position of the hot carrier. By default this last effect

is disabled but may be enabled with the parameters E.BENDING and H.BENDING for electrons and holes respectively.

The carrier velocity model follows the approach of Fiegna [129] where velocity is proportional to energy according to:

$$v_{\perp} \sim \varepsilon^{0.25} \qquad 3\text{-}294$$

The function $F(\varepsilon, T_{n,p}(x,y))$ is determined by the density of states and the energy distribution function according to:

$$F(\varepsilon, T_n(x,y)) \sim \frac{g(\varepsilon)f(\varepsilon)}{\int_0^{\infty} g(\varepsilon)f(\varepsilon)d\varepsilon} \qquad 3\text{-}295$$

The density of states $g(\varepsilon)$ follows the analysis of Cassi [141] where:

$$g(\varepsilon) \sim \varepsilon^{0.25} \qquad 3\text{-}296$$

Finally the energy distribution functions for electrons and holes are defined by:

$$f_e(\varepsilon) \sim \left[\exp\left(\frac{-\mathbf{ETH.N}\ \varepsilon^3}{T_n^{1.5}}\right) + \mathbf{C0}\exp\left(\frac{-\mathbf{ETH.P}\ \varepsilon^3}{T_n^{1.5}}\right)\right] \qquad 3\text{-}297$$

$$f_h(\varepsilon) \sim \exp\left(\frac{-\mathbf{CHI.A}\ \varepsilon^3}{T_p^{1.5}}\right) \qquad 3\text{-}298$$

where ETH.N, ETH.P, CHI.A and C0 are user definable constants found from fitting to measured data. The terms T_n and T_p are the mean carrier temperatures for electrons and holes which are calculated from the energy balance model.

Normalization in all of the above equations is accounted for in the combined constants of proportionality, CGATE.N and CGATE.P.

The second probability P_1 is the probability that no energy is lost by optical phonon scattering as the hot carrier travels towards the semiconductor-insulator interface after being redirected, and is given by:

$$P_{1,n} \sim \exp\left(-\frac{r}{\mathbf{IG.ELINF}}\right) \qquad 3\text{-}299$$

$$P_{1,p} \sim \exp\left(-\frac{r}{\mathbf{IG.HLINF}}\right) \qquad 3\text{-}300$$

where r is the distance from point of redirection to the semiconductor-insulator interface.

The final probability P_2 accounts for the probability of scattering in the image force potential well in the gate oxide and is given by;

$$P_{2,n} = \exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{\mathbf{PATH.N}}\right) \quad \text{for } \theta > \mathbf{THETA.N} \qquad 3\text{-}301$$

$$P_{2,n} = 0 \quad \text{for } \theta > \mathbf{THETA.N} \qquad 3\text{-}302$$

$$P_{2,p} = \exp\left(-\frac{\sqrt{\frac{q}{16\pi\varepsilon_{ox}E_{ox}}}}{\mathbf{PATH.P}}\right) \quad \text{for } \theta > \mathbf{THETA.P} \qquad 3\text{-}303$$

$$P_{2,p} = 0 \quad \text{for } \theta < \mathbf{THETA.P} \qquad 3\text{-}304$$

where PATH.N and PATH.P are the electon and hole mean free path lengths within the oxide, ε_{ox} is the oxide permittivity and E_{ox} is the electric field in the oxide. The angle θ introduces an angle dependence which is based upon the work of Wada [131]. His experiments indicate a critical rejection angle, THETA.N and THETA.P, between the angle θ formed between the semiconductor-insulator interface and the electric field in the oxide. If the angle θ is less than the rejection angle thenthe electrons are repelled back to the substrate.

Note: The current implementation of the Concannon model for hot carrier injection is that only carriers along the semiconductor-insulator interface are significant and as a result the probability P_1 is assumed unity. This also means that the integration is only applied to those node points along the semiconductor-insulator interface.

Two other parameters of the MODELS statement that may affect the result of the numeric integration are user definable. The ENERGY.STEP parameter specifies the energy step size in eV used during the numeric integration. The default step size is 25 meV. The INFINITY parameter sets the upper limit of the integration and specifies ratio of the increment added to the integral divided by the current value of the integral. The default value of the INFINITY parameter is 0.001.

The implementation of this model is similiar to that for Fowler-Nordheim tunneling. Each electrode-insulator and insulator-semiconductor interface is divided into discrete segments which are based upon the mesh. For each insulator-semiconductor segment the Fowler-Nordheim current is calculated as described above. This current will then be added to a segment on the electrode-insulator boundary. Two schemes have been implemented to find out to which segment this current should be added.

The default model that calculates which electrode segment receives the hot carrier injected current follows the path of the electric field vector at the semiconductor-insulator interface. The first electrode-insulator segment that is found along this trajectory, provided no other semiconductors or metals are found along the trajectory, will receive the current.

A second model may be chosen using the NEARFLG parameter of the MODELS statement. In this case the electrode-insulator segment found closest to the semiconductor-insulator segment will receive the hot carrier injected current.

The total current on the gate electrode is then the sum of the currents from all the individual segements around the electrode boundary.

Note: To maintain self-consistent results it is important that this model is implemented if the Concannon model is being used for the simulation of substrate current.

Device Level Reliability Modeling

Hansch MOS Reliability Model

The Hansch Reliability Model [88] can be used to simulate MOS transistor degradation under stress conditions. The causes of device characteristic degradation are the hot electron (hole) injection into gate oxide, and the trapping of electron (hole) charge on the effective interface acceptor (donor) like traps.

The model calculates hot electron (hole) injection current according to the lucky electron model. The arbitrary position-dependent distributions of acceptor and donor-like traps are specified on the oxide-semiconductor interface as *a priori* knowns, with corresponding capture cross sections. The device degradation is calculated as a function of stress time by performing transient calculations. The trap rate equation is solved on every time-step and thus the trapped electron (hole) concentration is calculated. The rate of electron trapping can be described by the equations:

$$\frac{dN_n(x,t)}{dt} = \frac{\mathbf{SIGMAE}}{q} \cdot J_{inj,\,n}(x,t) \cdot (\mathbf{NTA}(x) - N(x,t)) \qquad 3\text{-}305$$

$$\frac{dN(x,t)}{dt} = \frac{\mathbf{SIGMAH}}{q} \cdot J_{inj,\,p}(x,t) \cdot (\mathbf{NTD}(x) - N(x,t)) \qquad 3\text{-}306$$

where N(x,t) represents the **trapped** electron (hole) density, at the interface point x, at **time=t** during a transient simulation. The NTA and NTD parameters represent the acceptor and donor-like trap densities at **time=0**. The $J_{inj,n}(x,t)$ and $J_{inj,p}(x,t)$ parameters are the injected electron and hole current densities, SIGMAE and SIGMAH are the capture cross section of electrons and holes.

To activate this model the parameters DEVDEG, DEVDEG.E, and DEVDEG.H may be used in the MODELS statement (to account for both hot electron and hole injection, hot electron or hot hole injection, respectively). The model parameters are user definable on the DEGRADATION statement.

Table 3-51. User Definable Parameters for Equations 3-305 and 3-306

Statement	Parameter	Units
DEGRADATION	SIGMAE	cm^2
DEGRADATION	SIGMAH	cm^2
DEGRADATION	NTA/F.NTA	cm^2
DEGRADATION	NTD/F.NTD	cm^2

The results of stress simulation can be used to calculate the characteristics of the degraded device (the shift of the threshold voltage, transconductance degradation, etc.). The distribution of traps, hot electron (hole) current density, and trapped electron (hole) distribution can be easily visualized using TONYPLOT.

The model parameters NTA, NTD, SIGMAE, and SIGMAH may also be defined through the C-interpreter functions F.NTA, F.NTD, F.SIGMAE, and F.SIGMAH. This allows these values to be defined as functions of their position (x,y) along the insulator-semiconductor interface. These C-function libraries are also defined on the DEGRADATION statement. More information on the C-interpreter functions can be found in Appendix A.

ORCHID

The Orchid module in ATLAS allows users to simulate total radiation dose effects on reliability in MOS devices. These effects are treated by modelling the gate oxide as a semiconductor. The user does this by adding a MATERIAL statement identifying the gate oxide region using the REGION parameter and adding the parameter SEMICONDUCTOR to the statement. This MATERIAL statement should immediately follow the structure definition or MESH statement if the structure is read in.

Next, the user must specify a C interpreter function, F.OXGENERATE, on the BEAM statement. This function specifies the generation rate of electron-hole pairs in the oxide as a function of position, time and electric field.

The user must also specify a C interpreter function, F.OXCHARGE, on the DOPING statement. This function specifies the oxide charge density as a function of position, field, electron and hole currents and time.

The user then, runs the simulation in the time domain, capturing the state of the structure at various times of interest. These structures can then be simulated to extract the ID-VG characteristics to examine the threshold shifts as a function of exposure.

The Ferroelectric Permittivity Model

Ferroelectric materials exhibit high dielectric constants, polarization and hysterisis. Such materials are finding more and more applications in integrated memory devices. To allow simulation of these effects a modified version of the ferroelectric model from Miller [113] has been inmplemented.

The ferroelectric model is enabled by setting the FERRO parameter of the MODELS statement. In this model the permitivity used in Poisson's equation (Eq. 3-1) is given the following functional form:

$$\varepsilon(E) = \mathbf{FERRO.EPSF} + \mathbf{FERRO.PS} \bullet 2\delta \bullet \operatorname{sech}^2\left[\frac{E - \mathbf{FERRO.EC}}{2\delta}\right] \qquad 3\text{-}307$$

where FERRO.EPSF is the permitivity, E is the electric field and δ is given as follows:

$$\delta = \mathbf{FERRO.EC}\left[\log\frac{1 + \mathbf{FERRO.PR}/\mathbf{FERRO.PS}}{1 + \mathbf{FERRO.PR}/\mathbf{FERRO.PS}}\right]^{-1} \qquad 3\text{-}308$$

The user definable parameters FERRO.EPSF, FERRO.PS, FERRO.PR, and FERRO.EC can be modified on the MATERIAL statement as shown Table 3-52.

The derivative of the dipole polarization with respect to electric field is given by:

$$\frac{dP_d}{dE} = \Gamma\frac{dP_{sat}}{dE} \qquad 3\text{-}309$$

where P_d is the position dependent dipole polarization. A numeric integration of this function is carried out in ATLAS to determine the position dependent dipole polarization.

For saturated loop polarization, the function Γ is equal to unity which corresponds to the default model. If the user specifies the parameter UNSAT.FERRO on the MODELS statement, the function Γ can take on a more general form suitable for simulation of unsaturated loops. In this case, the function Γ is given by:

$$\Gamma = 1 - \tanh\left[\left(\frac{P_d - P_{sat}}{\xi P_s - P_d}\right)^{1/2}\right] \quad 3\text{-}310$$

where ξ =1 for increasing fields and ξ = -1 for decreasing fields.

Table 3-52. User Specifiable Parameters for Equations 3-307 to 3-308.

Statement	Parameter	Default	Units
MATERIAL	FERRO.EC	0.0	V/cm
MATERIAL	FERRO.EPS	1.0	
MATERIAL	FERRO.PS	0.0	C/sqcm
MATERIAL	FERRO.PR	0.0	C/sqcm

Quantum Mechanical Models

Self Consistent Schrodinger Poisson Model

To model the effects of quantum confinement, ATLAS also allows solution of Schrodinger's equation along with the fundimental device equations. The solution of Schrodinger's equation gives a "quantized" description of the density of states in the presence of quantum mechanical confining potential variations.

The calculation of the "quantized" density of states relies upon a solution of Schrodinger's equation:

$$\nabla\left(\frac{1}{m_e}\nabla\zeta_k\right) + U_{tot}\zeta_k = \lambda_k\zeta_k \quad 3\text{-}311$$

where ζ_k and λ_k are the k^{th} bound state wavefunction and energy level respectively and U_{tot} is the total potential energy of the electron.

$$U_{tot} = qV + E_h \quad 3\text{-}312$$

where E_h, the conduction band heterojunction discontinuity and is typically given by the difference in electron affinities:

$$E_h = \chi|_{top} - \chi|_{botton} \quad 3\text{-}313$$

The quantized density of states subsequently has the form:

$$g(E)_{quantum} = \left.\frac{4\pi m_e}{h^2} \cdot \left|\zeta_k\right|^2\right|_{E = \lambda_k} \qquad 0\big|_{E \neq \lambda_k} \qquad \text{3-314}$$

Using Fermi-Dirac statistics, the discrete nature of the quantised density of states reduces the integral over energy to a sum over bound state energies:

$$n = N_{C_{2D}} \sum_{k=1}^{k_{MAX}} \left|\zeta_k\right|^2 \log\left(1 + e^{\left(\frac{E_F - \lambda_k}{k_B T}\right)}\right) \qquad \text{3-315}$$

where $N_{C_{2D}}$ is also a constant for a given effective mass and temperature, given by the Equation:

$$N_{C_{2D}} = 2\left(\frac{2\pi m_e k_B T}{h^2}\right) \qquad \text{3-316}$$

The self-consistent Schrodinger-Poisson model is enabled by setting the SCHRO parameter of the MODEL statement. With this parameter set ATLAS solves the one dimensional Schrodinger's equation along a series of slices in the y direction relative to the device. Each slice is taken along an existing set of Y nodes in the ATLAS device mesh. Note: This feature can only be used with rectangular grids in ATLAS.

After the Schrodinger equation solution is taken, carrier concentrations calculated from equation 3-315 are substituted into the charge part of Poisson's equation. The potential drived from solution of Poisson's equation is substituted back into Schrodinger's equation. This solution process (alternating between Schrodinger's and Poisson's equation) is continued until convergence is reached and a self-consistent solution of Schrodinger's and Poisson's equation is obtained. The solutions of the self consistent system can be written to a structure file using the SAVE statement or using the OUTFILE parameter of the SOLVE statement. These structure files will contain the self-consistent potential and electron concentrations. The Eigen energies and Eigen functions can also be written to the structure file by specifying the EIGENS parameter of the OUTPUT statement. This parameter specifies the number of Eigen energies/wavefunctions to be written. Note: The number of Eigen values solved is limited to a number of 2 less than the total number of grid points in the Y direction. It should be noted that the self-consistent solution of Schrodinger's equation with Poisson's equation does not allow for solutions of the electron and hole continuity equations in the current ATLAS version. However, non-self-consistent solutions can be obtained by setting the POST.SCHRO parameter of the MODELS statement. These non-self-consistent solutions are obtained by solving Schrodinger's equation only after convergence is obtained. In this manner Schrodinger soltuions can be obtained with the electron and hole continuity equations. Similar results are saved to the structure file and the meaning of the EIGENS parameter of the OUTPUT statement are the same. In obtaining post-processed (POST.SCHRO) solutions to Schrodinger's equation some assumption is made about the location of the electron quasi-fermi level. Two flags can be set on the MODELS statement to vary the interpretation of the results. These parameters are:

FIXED.FERMI and CALC.FERMI

and have interpretations as outlined in Table 3-53.

Table 3-53. Interpretations of FIXED.FERMI and CALC.FERMI Parameters during Post-Processed Schrodinger solution		
FALSE	FALSE	Quasi-Fermi level is calculated from the local electron density via Equation 3-9.
FALSE	TRUE	Quasi-Fermi level is uniform across Y slice and is calculatd to match the classical and quantum-mechanical sheet charge.
TRUE	FALSE	Quasi-Fermi level is uniformly zero.
TRUE	TRUE	Quasi-Fermi level varies with Y position and is calculated to match the local classical and quantum-mechanical charge concentration.

Quantum Moments Model

The quantum module in ATLAS is applicable to several different types of problems:

1. HEMT channel confinement simulation
2. Thin gate oxide MOS capacitors and transistors
3. Other problems such as small geometry MESFETs and Heterojunction diode

The effects due to comfinement of carriers associated with variations of local potential on the scale of the electron wave functions (i.e., quantum effects) can be modeled in ATLAS using a quantum transport model. This model is based on moments of the Wigner function equations-of-motion [120, 121] consists of quantum correction to the carrier temperatures in the carrier current and energy flux equations (Equations 3-74, 3-75, 3-77 and 3-78). The quantum correction to the carrier temperature is given by:

$$T_q = T + \frac{2}{3k_b} U_q \qquad 3\text{-}317$$

where T*q* is the quantum corrected carrier temperature, T is the carrier temperature and U*q* is the quantum potential. When the quantum moment model is enabled the carrier temperatures appearing in equations 3-74, 3-75, 3-77 and 3-78 are replaced by the quantum corrected temperature Tq.

The quantum potential, U*q* is given by:

$$U_q = -\frac{h^2}{8m^*} \nabla^2 \ln(n) \qquad 3\text{-}318$$

where *n* is the carrier concentration.

Equations 3-317 and 3-318 apply to both electrons and holes. In ATLAS, the quantum transport model is enabled for electrons by specifying E.QUANTUM on the MODEL statement, and is enabled for holes by specifying H.QUANTUM on the MODEL statement.

The quantum temperatures can be written to the standard structure file by specifying T.QUANTUM on the OUTPUT statement. Once written the quantum temperature distribution can be examined using TONYPLOT.

Note: In equating Equation 3-318, the quantum potential is proportional to the gradient squared of the log of the carrier concentration, the quantum temperature tends to increase the thermal diffusivity around sharp peaks in the concentration distribution, and tends to smear out such peaks. This effect tends to draw carriers away from the Si-Si02 interface in MOSFETS. In HEMTS, it helps move carriers away from the hetero-interface and into the narrow bandgap channel.

Syntax

As discussed in the previous section, the quantum model uses a quantum moments function added to the standard drift diffusion current continuity equation. As such, the model for electrons and holes can be activated independently. The quantum model is activated on the MODELS statement using the quantum switch:

```
models quantum
```

will activate the quantum moments equation for electrons, and

```
models h.quantum
```

will activate the quantum moments equation for holes.

Since the distributions of carriers given by the quantum moments model can vary significantly from the distributions predicted by the standard drift-diffusion or energy-balance models, the standard initial guess strategies (e.g. INIT) typically, are not suitable for obtaining solutions for quantum moments. Until more suitable initial guess strategies can be devised for quantum moment modelling, we have included a damping factor to gradually apply the quantum moment model. This damping factor is specified by the QFACTOR parameter of the SOLVE statement. The QFACTOR is implemented as a prefactor to the expression for the quantum temperature in equation 3-17. As such a value of QFACTOR of 0.0 implies that the quantum moment model is turned off or is not being applied. A value of QFACTOR of 1.0 implies that the quantum moment model is turned on and is completely applied. The user may smoothly vary the value of QFACTOR between 0.0 and 1.0 to overcome the problems of initial guess. During this ramping of QFACTOR, PREVIOUS should be used as an initial guess. Also, during such variation of QFACTOR, trapping is available, should a given solution not converge. In such cases, the QFACTOR is reduced, and solutions proceed until convergence is obtained or the maximum number of trap steps is exceeded.

The following is a typical fragment from an input file that ramps the QFACTOR parameter from 0 to 1:

```
model numcarr=1 quantum fldmob conmob srh print
output con.band val.band band.param t.quantum
solve init
solve qfactor=0.0
solve qfactor=0.0001
solve qfactor=0.001
solve qfactor=0.01
solve qfactor=0.1
solve qfactor=1.0
log outf=test.log
solve vdrain=0.01
```

Note: A `t.quantum` switch on the output statement includes the 'quantum temperature' across the semiconductor, as discussed in the previous section.

Examples

An example of the application of this model would be the quantization of the electrons in an AlGaAs/GaAs `HEMT` structure. The figures below illustrates such a structure, as well as an outline through the channel for both a classical type structure and a quantum structure illustrating the concentration of electrons throughout the device.

Note: The 'smoothing out' of the sharp carrier peaks predicted by the quantum model which is characteristic of a quantized channel device.

Quantum Correction Models

Hansch's Model

The quantum mechanical correction as given by Hansch et. al. [124] is suitable for accurate simulation of the effects of quantum mechanical confinement near the gate oxide interface in MOSFETs. This correction factor is essentially a modification of the density of states as a function of depth below the Si/SiO_2 interface and is given by Equation 3-319.

$$N_C^* = N_C\left(1 - e^{-\frac{z}{LAMBDA^2}}\right)^2 \qquad 3\text{-}319$$

Here N_C is the standard density of states, z is the depth below the interface and LAMBDA is a user specifiable parameter on the `MODEL` statement.

Van Dort's Model

In deep submicron MOS devices, quantum effects in the channel can have significant effects on the device characteristics. These effects are directly due to the increased doping levels and thinner gate oxide. In the channel of such devices, the potential well well formed during inversion is of dimensions where the effects of quantum confinement must be considered. The direct effect of such quantum confinement is that the peak of the carrier concentration is shifted away from the interface and the thinner gate oxide can cause a marked difference in gate capacitance. Consideration of the effects may be essential for accurate prediction of the device turn-on voltage.

A full treatment of the quantum effects by solving, for example, Schrodinger's equation, is not always desirable from a computational standpoint, nor necessary. A simpler approach has been suggested by Van Dort [123]. In this model, the effects of the quantum confinement are modeled by an effective broadening of the bandgap near the surface as a function of a perpendicular electric field and distance from the surface. In Van Dort's model, the change in bandgap is given by the expression in equation 3-320.

$$\Delta E_g = B \cdot DORT\ \beta\frac{13}{9}\left(\frac{\varepsilon_{si}}{4_q k_B T_L}\right)^{1/3} (E_\perp)^{2/3} g(y) \qquad 3\text{-}320$$

Here, B.DORT is a user definable adjustment parameter on the MODEL statement, β is equal to 6.1×10^{-8} eV cm, $E_{\perp}$ is the perpendicular electric field and g(y) is a function to restrict the application of the model to the channel region, given by equatioin 3-321.

$$g(y) = \frac{2e^{-(Y/D \cdot DORT)^2}}{1 + e^{-2(Y/D \cdot DORT)^2}} \qquad 3\text{-}321$$

The Van Dort model for N channel devices is enabled by specifying N.DORT on the MODELS statement.

Chapter 4: S-PISCES

Introduction

Note: Before reading this chapter you should be familiar with ATLAS. If you are not familiar with basic operation of ATLAS, read the "Getting Started" chapter.

S-PISCES is a powerful and accurate two-dimensional device modeling program that simulates the electrical characteristics of silicon based semiconductor devices including MOS, bipolar, SOI, EEPROM and power device technologies.

S-PISCES calculates the internal distributions of physical parameters and predicts the electrical behavior of devices under either steady-state, transient, or small signal AC conditions. This is performed by solving Poisson's equation and the electron and hole carrier continuity equations in two dimensions.

S-PISCES solves basic semiconductor equations on non-uniform triangular grids. The structure of the simulated device can be completely arbitrary. Doping profiles and the structure of the device may be obtained from analytical functions, experimentally measured data, or from process modeling programs SSUPREM3 and ATHENA.

Simulating Silicon Devices Using S-PISCES

Simulating MOS Technologies

Physical Models for MOSFETs

S-PISCES provides special physical models tuned for simulating MOS devices. Most of these models are accessed from the MODEL statement. The MOS parameter of the MODEL statement can be specified to turn on a default set of physical models that are most useful for MOS simulation. The MOS parameter enables Shockley-Read-Hall (SRH), Fermi statistics (FERMI) and the Lombardi mobility model (CVT) for transverse field dependence. To sets the default MOS simulation models use.

```
MODEL MOS PRINT
```

The transverse field dependent mobility models are of particular importance for simulating MOS devices. S-PISCES currently supports several different transverse field dependent mobility models. The parameter CVT selects the Lombardi CVT model. The parameter YAMA selects the Yamaguchi model. The parameter TASCH selects the Tasch model, and the parameter WATT selects the Watt surface model which can be operated in a more accurarate mode with the extra parameter MOD.WATT on the MOBILITY statement.. Advanced users will find that the MOBILITY statement can be used to modify some of the parameters of the various models, to apply different models to different regions or to apply different models to electrons and holes.

Meshing for MOS Devices

In device simulation of MOS devices the key areas for a tight mesh are:

1. very small vertical mesh spacing in the channel under the gate. The exact size of mesh required depends on the transverse field or surface mobility model chosen. See figure 4-1 for an example of the effect of mesh size on drain current.
2. lateral mesh spacing along the length of the channel for deep sub-micron devices. This is required to get the correct source-drain resistance and to resolve the channel pinch-off point.

3. lateral mesh at the drain/channel junction for breakdown simulations. This is required to resolve the peak of electric field, carrier temperature and impact ionization.
4. several vertical grid spacings inside the gate oxide when simulating gate field effects such as gate induced drain leakage (GIDL) or using any hot electron or tunneling gate current models

The figures 4-1 and 4-2 show the effect of mesh size in the MOS channel on IV curves. In Figure 4-1 the mesh density in the vertical direction is increased. As the mesh density increases the resolution of the electric field and carrier concentration is improved. This example uses the CVT mobility model. Improvements in transverse electric field resolution lead to a reduced mobility in the channel and a stronger IV rolloff.

However Figure 4-2 shows the effect of surface channel mesh in MOSFETs is model dependent. This result shows the current at Vds=3.0V and Vgs=0.1V versus the size of the first grid division into the silicon. Results vary for each model but note that for all models a very fine grid is required in order to reduce the grid dependence to acceptable levels.

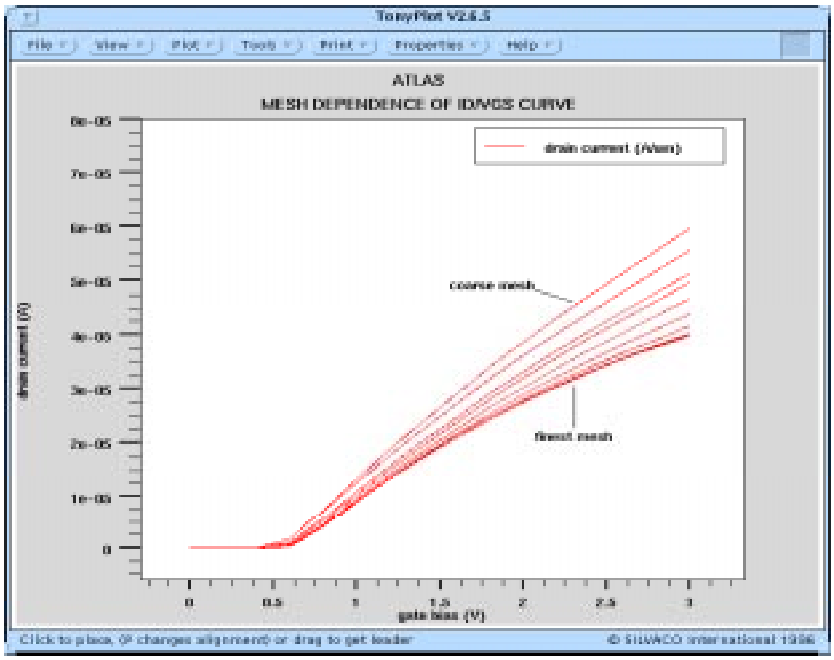

Figure 4-1: Effect on MOS IV curve of progressive refinement of the vertical mesh spacing at the surface of the MOS channel

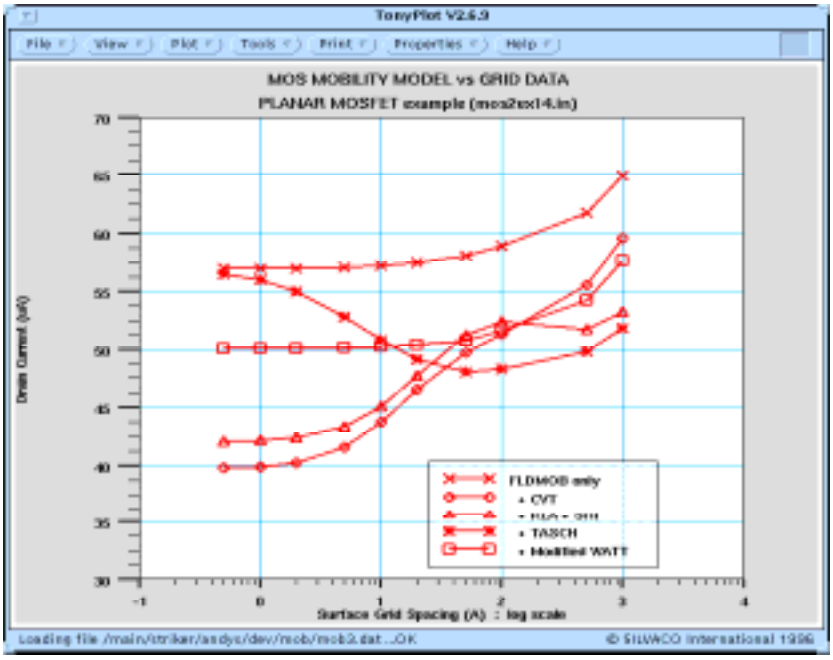

Figure 4-2: Effect of surface mesh spacing on simulated current for several MOS Mobility Models

MOS Electrode Naming

For MOS simulation, S-PISCES allows the use of standard electrode names to reduce confusion with the use of electrode indices. These names include "source", "drain", "gate", and "substrate". Electrode names can be defined in ATHENA or DEVEDIT or in the ELECTRODE statement in ATLAS. These names can be used in the SOLVE statements for setting bias voltages such as:

```
SOLVE VGATE=1.0 VSTEP=1.0 VFINAL=5.0 NAME=GATE
```

Gate Workfunction

In MOS simulations the workfunction of the gate is an important parameter. This must be set in each input deck using the WORK parameter of the CONTACT statement. For example:

```
CONTACT NAME=GATE WORK=4.17
```

would set the workfunction on the gate at 4.17 eV.

Certain material names can also be used to set the workfunction of common gate materials. For example:

```
CONTACT NAME=GATE N.POLY
```

would set the workfunction on the gate to that of n type polysilicon.

Note: The gate workfunction should be set on a CONTACT statement even though the material or workfunction might be set from ATHENA or DEVEDIT.

Interface Charge

For accurate simulation of MOS devices the interface charge at the oxide - semiconductor interface should be specified. This can be done by setting the QF parameters for the INTERFACE statement. Typically, a value of 3e10 is representative for the interface charge found in silicon MOS devices. The proper syntax for setting this value for interface fixed charge is:

```
INTERFACE QF=3e10
```

The user may also choose to try to model the fixed charge more directly by using interface traps to simulate the surface states. This can be done using the INTTRAP statement. This is almost never done in practice.

Single Carrier Solutions

Frequently, for MOS simulation the user may choose to simulate only the majority carrier. This will significantly speed up simulations where minority carrier effects can be neglected. This can be done by turning off the minority carrier. This is done using the ATLAS negation character "^" and one of the carrier parameters, ELECTRONS or HOLES, on the METHOD statement. For example, if the the user wanted to simulate electrons only, you could specify:

```
METHOD CARRIERS=1 ELECTRONS
```

or

```
METHOD ^HOLES
```

Single carrier solutions should not be performed where impact ionization, any recombination mechanism, lumped element boundaries, or small signal AC analysis are involved.

Energy Balance Solutions

As MOS devices become smaller and smaller, non-local carrier heating effects become important. Accurate simulation of non-local carrier heating effects can be simulated using the energy balance model (EBM). As a general rule the gate length can be used as a gauge to predict when non-local effects are important. Generally, for drain currents energy balance should be applied for gate lengths less than 0.5 microns, for substrate currents energy balance should be applied for gate lengths less than 1.0 micron.

To enable energy balance for electrons/holes, the HCTE.EL or HCTE.HO parameters should be set on the MODEL statement. For example:

```
MODEL HCTE.EL
```

enables the energy balance model for electrons.

ESD Simulation

In some cases lattice heating may be of importance for MOS simulation. This typically occurs in cases with very high currents, as is the case with ESD simulation. In these cases GIGA should be used to simulate the heatflow in the device. To enable heat flow simulation, the LAT.TEMP parameter of the MODEL statement should be set (a license for GIGA is required). For example, the statement:

```
MODEL LAT.TEMP
```

enables heatflow simulation.

Simulating Silicon Bipolar Devices

Physical Models for BJTs

S-PISCES provides special physical models for bipolar device simulation. These models can be selected using the MODEL statement. The BIPOLAR parameter of the MODEL statement enables a reasonable default set of bipolar models. These include: concentration dependent mobility (CONMOB), field dependent mobility (FLDMOB), band-gap narrowing (BGN), concentration dependent lifetime (CONSRH) and Auger recombination (AUGER).

For the most accurate bipolar simulations the recommended mobility model is the Klaassen Model (KLA). This includes doping, temperature and carrier dependence. It applies separate mobility expressions for majority and minority carriers. This model should also be used with Klaasen's Auger model (KLAAUG) and Klaassen's concentration dependent SRH model (KLASRH). The mobility model should be combined with FLDMOB to model velocity saturation. For surface (or lateral) bipolar devices the Shirahata model (SHI) can be used to extend the Klaassen model with a transverse electric field dependence. The most accurate and appropriate model statement for bipolar devices is therefore:

```
MODELS KLA FLDMOB KLASRH KLAAUGER BGN FERMI PRINT
```

This set of models may be chosen with the parameter BIPOLAR2 of the MODEL statement.

Note: For a complete syntax including description of models and method for simulating polysilicon emitter bipolar devices see the BJT directory in the on-line examples

---

Meshing Issues for BJTs

The most important areas to resolve in bipolar transistors are the emitter/base and base/collector junctions. Typically a very fine mesh throughout the base region is required. The gain of the bipolar device is determined primarily by the recombination at the emitter/base junction or inside the emitter. Hence these regions need to be resolved with a fine mesh.

BJT Electrode Naming

S-PISCES also provides special electrode names for bipolar simulation that can be used to ease confusion over electrode indices. These electrode names include: "emitter", "base", "collector", "anode", and "cathode". Electrode names can be defined in ATHENA or DEVEDIT or in the ELECTRODE statement in ATLAS. The electrode names are used on SOLVE statement for example:

```
SOLVE VBASE=1.0 VSTEP=1.0 VFINAL=5.0 NAME=BASE
```

Dual Base BJTs

It is possible in S-PISCES to tie two or more contact together so that voltages on both contacts are equal. This is useful for many technologies for example dual base bipolar transistors. There are several methods for achieving this depending on how the structure was initial defined.

If the structure is defined using ATLAS syntax it is possible to have multiple ELECTRODE statements with the same NAME parameter defining separate locations within the device structure. In this case the areas defined to be electrodes will be considered as having the same applied voltage. A single current will appear combining the current through both ELECTRODE areas.

Similarly if two separate metal regions in ATHENA are defined using the ATHENA ELECTRODE statement to have the same name, then in ATLAS these two electrodes will be considered as shorted together.

If the electrodes are defined with different names the following syntax canbe used to link the voltages applied to the two electrodes.

```
CONTACT NAME=base1 COMMON=base
.
SOLVE VBASE=0.1
```

Here the electrode *base1* will be linked to the electrode *base*. Later the applied 0.1V on *base* will also appear on *base1*. However ATLAS will calculate and store **separate** currents for both *base* and *base1*. This can be a useful feature. However in some cases such as where functions of the currents are required in EXTRACT or TONYPLOT it is undesirable. The parameter SHORT may be added to the CONTACT statement above to specify that only a single *base* current will appear combining the currents from *base* and *base1*.

When loading a structure from ATHENA or DEVEDIT where two defined electrode regions are touching, ATLAS will automatically short these and use the electrode name that was defined first.

Creating an Open Circuit Electrode

It is often required to perform a simulation with an open circuit, such as for an open-base breakdown voltage simulation, on one of the defined electrodes. From a device simulation viewpoint there are three different methods that will accomplish this. These are;

- Entirely deleting an electrode from the structure file.
- Adding an extremely large lumped resistance, for example $10^{20}\Omega$, onto the contact to be made open circuited.
- Switching the boundary conditions on the contact to be made open circuited from voltage controlled to current controlled and then specifying a very small current through that electrode.

Each of these methods are feasible but if in doing so a floating region is created within the structure then numerical convergence may be affected. As a result it is normally recommended that the second method, that of a lumped resistance, be used as it ensures that no floating region is created.

Solution Techniques for BJTs

To obtain bipolar solutions it is almost always necessary to simulate using two carriers. This is due to the importance of minority carriers to device operation.

In certain cases non-local carrier heating may be of importance for the accurate simulation of bipolar devices. In these cases the energy balance model should be used. To model non-local carrier heating for electrons/holes, the HCTE.EL, HCTE.HO parameters should be set on the MODEL statement. For example, the statement:

```
MODEL HCTE.EL
```

invokes the carrier heating equation for electrons.

Simulating Non-Volatile Memory Technologies (EEPROMs, FLASH Memories)

As might be expected users wishing to simulate non-volatile memory devices should first become familiar with the basics of MOSFET simulation described earlier in this chapter

Defining Floating Gates

To simulate non-volatile memory technologies such as EEPROMs or FLASH EEPROMs it is necessary to specify one or more electrodes as floating gates. This is done by setting the FLOATING parameter of the CONTACT statement. For example:

```
CONTACT NAME=fgate FLOATING
```

This specifies that the electrode named "fgate" is simulated as a floating gate. This means that the charge on the floating gate is calculated as the integral of the gate current at that gate during a transient simulation.

Modeling the correct coupling capacitance ratio between the floating gate and control gate often requires adding an extra lumped capacitor from the floating gate to the control gate or one of the other device terminals. This is often required since S-PISCES is performing a 2D simulation whereas the coupling of the gates is often determined by their 3D geometry. Parameters on the CONTACT statement are used to apply these extra lumped capacitances. For example to add a capacitor of 1fF/μm between the control and floating gates the syntax is:

```
CONTACT NAME=fgate FLOATING FG1.CAP=1.0e-15 EL1.CAP=cgate
```

Gate Current Models

The gate currents for the floating gate structure can be supplied by one of three sources: hot electron injection (HEI or N.CONCAN) , hot hole injection (HHI or P.CONCAN) and Fowler-Nordheim tunneling current (FNORD).

These currents are of importance depending on whether electrons are being moved onto the gate or off the floating gate. In the case of placing electrons on the floating gate hot electron injection and Fowler-Nordheim tunneling should be used. In the case of removal of electrons from the floating gate hot hole injection and Fowler-Nordheim tunneling should be set.

In drift diffusion simulations hot electron injection is simulated by setting the HEI parameter of the MODELS statement. Hot hole injection is simulated using the HHI parameter of the MODELS statement. Fowler-Nordheim tunneling is enabled by setting the FNORD parameter of the MODELS statement. The following example demonstrated the proper setting for Flash EPROM programming:

```
MODELS MOS HEI PRINT
```

The next example would be appropriate for EEPROM erasure:

```
MODELS MOS HHI FNORD PRINT
```

With energy balance simulations the Concannon Models should be used for EPROM programming and erasing.

Note: Writing and erasure of floating gate devices should be done using transient simulation.

Gate Current Assignment (NEARFLG)

The actual calculation of floating gate current magnitude is done at the silicon-oxide interface. The question of distribution of oxide currents to the various electrodes near the interface is resolved using one of two models. The actual flow of carriers in oxides is not well known. Accepted physical models of carrier flow in oxides are still under research. As such, S-PISCES provides two heuristic models to choose from. The default is distribute currents calculated at points along the interface to the electrode in the direction of highest contributing field. This model is somewhat analogous to a purely drift model of oxide carrier transport. The alternative is to set the NEARFLG parameter of the MODEL statement. In this case the currents calculated at points along the interface are distributed to the geometrically closest electrode. This model is analogous to a purely diffusion model of carrier transport in oxide.

Simulating SOI Technologies

Silicon substrates are now being produced that contain a thin oxide layer buried below the surface of the silicon at some predefined depth. The existence of this buried oxide layer has resulted in a change not only in the fabrication process used to manufacture a device in the surface silicon, but also in the challenges facing device simulation.

All of the issues raised previously about MOS device simulation should also be considered with some extra SOI specific problems.

The most common device technology that uses these SOI substrates is the SOI MOSFET. This section shall try to summarise the simulation requirements for SOI using this particular technology as a reference.

Meshing in SOI devices

The mesh requirements for SOI MOSFETs is very similiar to that described in the previous section for bulk MOS transistors. In addition to these four requirements there are some additional points

1. Two channel regions may exist; one underneath the top (front) gate oxide and one above the buried (back gate) oxide.
2. Inside the buried oxide layer the mesh constraints may be relaxed considerably compared with the top gate oxide.
3. The active silicon underneath the top gate may act as the base region of a bipolar transistor and as such may require a finer mesh when compared to bulk MOS transistors.

Physical Models for SOI

Simulation of SOI MOSFETs is based upon the physical operation of the device which exhibits both MOS and bipolar phenomena. As a result a more complex set of physical models will be required than for either MOS or bipolar technologies.

Mobility: Klaassens model (KLA) is recommended to account for lattice scattering, impurity scattering, carrier-carrier scattering and impurity clustering effects at high concentration. The Shirahata mobility model (SHI) is needed to take into account surface scattering effects at the silicon/oxide interface which is a function of the transverse electric field. High electric field velocity saturation is modelled through the field dependent mobility model (FLDMOB). Model parameters may be tuned using the MOBILTY statement syntax.

Interface Charge: In SOI transistors there exist two active silicon to oxide interfaces on the wafer. The top interface, under the top gate, is similiar to MOS technology. The bottom interface is quite different and typically contains significantly more charge. Different interface charges may be set in SPISCES using the INTERFACE statement with region specific parameters.

Recombination: To take account of recombination effects we recommend the use of the Shockley-Read-Hall (SRH) model. This will simulate the leakage currents that exist due to thermal generation. It may also be necessary to simulate the presence of interface traps at the silicon/oxide interface. Finally the direct recombination model (AUGER) should be turned on. The parameters for both models may be tuned on the MOBILITY statement.

Bandgap Narrowing: This model (BGN) is necessary to correctly model the bipolar current gain when the SOI MOSFET behaves like a bipolar transistor. The parameters for this model are specified on the MODELS statement.

Carrier Generation: Impact ionization significantly modifies the operation of SOI MOSFETs. To account for this phenomena the impact ionization model (IMPACT) should be switched on and calibrated for SOI technology. The calibration parameters are set on the IMPACT statement.

Lattice Heating: When a device is switched on there can be significant current density within the silicon. This could generate a significant amount of heat. In bulk MOS devices the silicon substrates behaves like a good heat conductor and this generated heat is quickly removed. However, this is not the case with SOI substrates as the buried oxide layer allows this generated heat to be retained. For SOI MOSFETs this can be a significant amount and can drastically affect the operation of the device. In such cases it would be necessary to take account for this by using the GIGA module. Note when lattice heating is switched on, by using the parameters LAT.TEMP on the MODELS statement, it is also necessary to specify a thermal boundary condition with the THERMCONTACT statement – see the GIGA section for more details.

Carrier Heating: In deep submicron designs it may also be important to switch on the additional *energy balance* equations. These take into account the exchange of energy between carriers and between the carriers and the lattice. See the section on MOS simulation for more information.

An example of a set of typical models for a partially depleted SOI MOSFET could be:

```
MODEL KLA SHI FLDMOB SRH AUGER BGN LAT.TEMP
INTERFACE QF=1e10 Y.MAX=0.05
INTERFACE QF=1e11 Y.MIN=0.05
THERMCONTACT NUM=1 ELEC.NUM=4 EXT.TEMP=300
IMPACT SELB
```

Numerical Methods for SOI

One important issue with SOI device simulation is the choice of numerical methods. In SOI technology the potential in the channel (or body) region is commonly referred to as "floating". This occurs because there exists no direct contact to it by any electrode. As a result when a bias is applied, or increased, on a contact there may be some convergence problem. This occurs because the guess used in the numerical solution scheme for (ψ, n ,p) may be poor, particularly in the "floating" region. This is particularly true if impact ionization is used. To overcome the problem of poor initial guess the following numerical methods syntax should be used in isothermal drift-diffusion simulations:

```
METHOD GUMMEL NEWTON
```

This method initially performs a Gummel iteration to obtain an improved initial guess for the Newton solution scheme. Although this method is more robust the penalty is that it has proved slower than using the Newton scheme alone. For more information on the numerical schemes available please refer to the Numerical Analysis section.

SOI Physical Phenomena

The physical models and the numerical schemes described above should allow S-PISCES to study all the important SOI phenomena. These include:

– full or partial depletion effects

– threshold voltage

– subthreshold slopes

– front to back gate coupling effects

– leakage current analysis

– high frequency analysis

– device breakdown

– snapback effects

– the "kink" effect in the output Ids-Vds characteristics

– negative differential resistance

[This page intentionally left blank]

Chapter 5: BLAZE

Introduction

Before continuing to the sections that follow, you should be familiar with ATLAS. If not, read Chapter 2 of this manual before proceeding with this chapter.

BLAZE is a general purpose 2-D device simulator for III-V, II-VI materials and devices with position dependent band structure (i.e. heterojunctions). BLAZE accounts for the effects of positionally dependent band structure by modifications to the charge transport equations. BLAZE is applicable to a broad range of devices including: HBTs, HEMTs, LEDs, heterojunction photodetectors (APDs, solar cells etc.), and heterojunction diodes.

This chapter is composed of several sections. The Basic Heterojuction Definitions section diagrams the basic heterojunction band parameters and includes a section on heterojunction alignment. Heterojunction Charge Transport is covered next and includes the details of how BLAZE modifies the basic transport models to simulate heterodevices. A section on the physical models unique to BLAZE is also included. Detailed information about the material systems encountered in heterojunction simulation is covered in the subsequent MATERIALS section. This includes the relationships bewteen the compound elemental concentrations and bandgap, dielectric constant, low field mobility and other important material and transport parameters. Defaults for these parameters can be found in the **A**ppendix B. Finally, the Simulating Heterojunction Devices with BLAZE section details a step by step approach to defining materials and models for simulation of heterojunction devices with BLAZE.

Basic Heterojunction Definitions

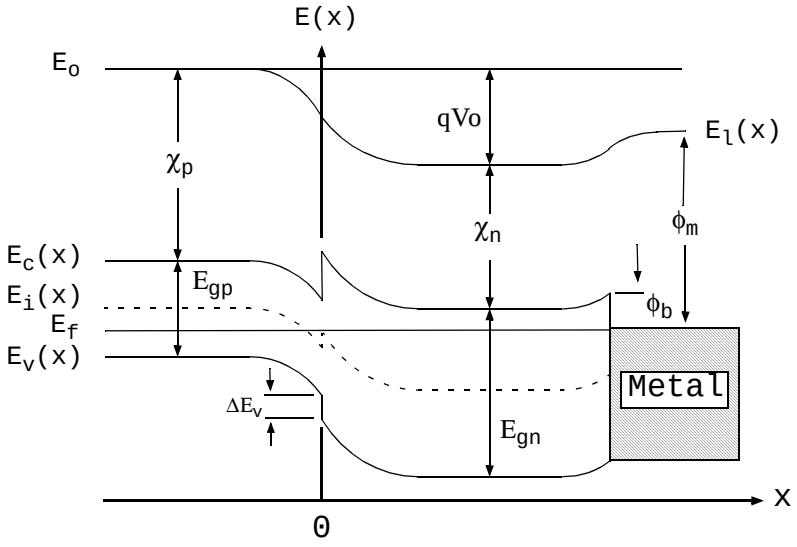

Figure 5-1: Band Diagram of p-n heterojunction

Figure 5-1 shows the band diagrams and band parameters for a basic p-n heterojunction device under equilibrium conditions. This diagram illustrates two materials with different bandgaps and electron affinities and a Schottky barrier in contact to the n-type material.

Referring to figure 5-1:

$E_c(x)$, $E_v(x)$ and $E_i(x)$ are the spacially dependent conduction band, valence band and intrinsic energy levels respectively.

E_f and E_o are the Fermi and Vacuum level energies.

E_{gp} and E_{gn} are the p-type and n-type material bandgaps.

ΔE_v is the portion of the difference between E_{gp} and E_{gn} that appears between the valence bands at the heterojunction interface.

ΔE_c (not labelled) is the portion of the difference between E_{gp} and E_{gn} that appears between the conduction bands at the heterojunction interface.

qVo is the built-in potential of the heterojunction.

χ_p and χ_n are the electron affinities of the p-type and n-type materials.

ϕ_m is the work function of the metal.

ϕ_b is the barrier height between the metal and the semiconductor.

The basic band parameters for defining heterojunctions in BLAZE are bandgap parameter **EG300**, electron parameter AFFINITY**,** and the conduction and valence band density of states **NC300** and **NV300.** These parameters are defined for each material using the MATERIAL statement. Other transport parameters relating these basic definitions to compound elemental concentrations (**x.composition** and **y.composition)** can also be defined. See the MATERIALS section of this chapter for a description of these relationships for each material system and the Simulating Heterojunction Devices with Blaze section for their usage. The work function of metals is defined using the CONTACT statement.

Alignment

As can be seen from figure 5-1, the difference in the two material bandgaps creates conduction and valence band discontinuities. How the bandgap difference is distributed between the conductance and valence bands has a large impact on the charge transport in these heterodevices. There are three methods for defining the conduction band alignment for a heterointerface: the affinity rule, the ALIGN parameter on the MATERIAL statement, and by manually adjusting the material affinities using the AFFINITY parameter on the MATERIAL statement.

The Affinity Rule

The default method in BLAZE for assigning how much of the bandgap difference appears as the conduction band discontinuity makes use of the affinity rule. The affinity rule assigns the conduction band discontinuity equal to the difference between the two materials electron affinities (AFFINITY on the MATERIAL statement). The affinity rule method is used by default for all materials where the ALIGN parameter has not been defined on the MATERIAL statement.

The ALIGN parameter on the MATERIAL statement

Experimental measurements of the actual band discontinuities can differ from what is assigned using the affinity rule with the standard material electron affinities. Therefore, BLAZE allows ΔE_c to be calculated by specifying the ALIGN parameter on the MATERIAL statement. ALIGN specifies the fraction of the bandgap difference which will appear as the conduction band discontinuity. This bandgap difference is between the material for which the ALIGN parameter is specified and the smallest bandgap material in the overall structure (the reference material). Internally, BLAZE creates the desired conduction band offset by modifying the electron affinity of the material for which the ALIGN parameter is specified.

In many applications Schottky barriers or more than two different semiconductor materials are present. The user must keep the reference material bandgap and these assigned affinities in mind when defining offsets for multiple materials or Schottky barrier heights. Examples for multiple materials and Schottky barriers are given in the examples section.

Manually Adjusting Material Affinity

The AFFINITY parameter on the MATERIAL statement can be used in conjunction with the default affinity rule alignment method to manually adjust the conduction band offset. In this case, the electron affinity of the larger bandgap material is adjusted so that the difference between the two materials affinity equals the desired conduction band offset. When more than two different materials are present, each material affinity can be adjusted in this manner. This is the easiest method for handling multiple materials and heterojunctions.

The following examples will describe the procedure for aligning heterojunctions using these three methods in BLAZE.

EXAMPLE 1:

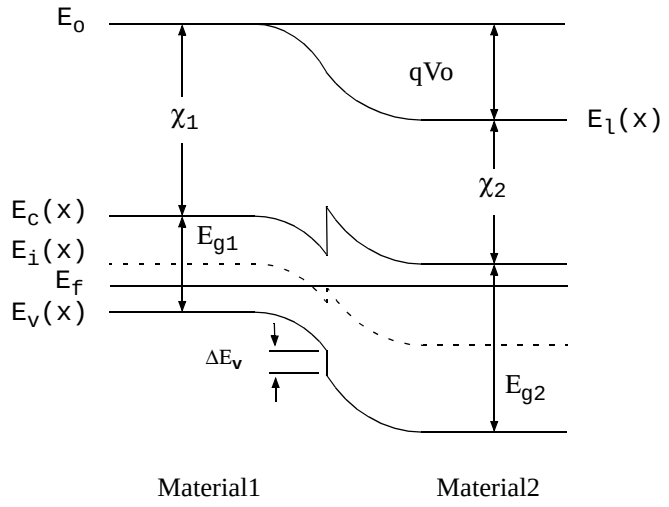

Figure 5-2: Band diagram of heterojunction with band offset.

Figure 5-2 diagrams a heterojunction consisting of two semiconductors with different bandgaps E_{g1} and E_{g2} and electron affinities χ_1 and χ_2. This example is similar to the bandstructure of a HFET or HEMT. For this example, $E_{g1} < E_{g2}$ and $\chi_2 < \chi_1$.

Allocating the conduction band offsets using the affinity rule:

$$\Delta E_c = \chi_1 - \chi_2 \qquad 5\text{-}1$$

and

$$\Delta E_v = \Delta E_g - \Delta E_c \qquad 5\text{-}2$$

ΔE_c is the amount of the conduction band discontinuity at the heterointerface and ΔE_v is is the amount of the valence band discontinuity.

Note: Remember the Affinity Rule is invoked to calculate the conduction band offset for a material as long as the ALIGN parameter is NOT specified on the MATERIAL statement for that material.

Using the ALIGN parameter on the MATERIAL statement

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset. Define the ALIGN parameter on the MATERIAL statement for Material 2 using

```
MATERIAL NAME=Material2 ALIGN=0.80
```

Then

$$\Delta E_c = (E_{g2} - E_{g1}) \cdot 0.80 \qquad 5\text{-}3$$

Internally, the affinity of Material 2 is adjusted so that ΔE_c equals this value. This value of electron affinity will override any electron affinity specification for Material 2. This has an impact on any calculation where this materials electron affinity is used and must be considered when specifying Schottky barriers contacted to this materials. See Example 4 for more details on Schottky barrier considerations

Manually Adjusting Material Affinity

```
MATERIAL NAME=Material2 AFFINITY=VALUE
```

where VALUE is adjusted to provide for the desired conduction band offset as calculated by the affinity rule, that is, relative to Material1. This value of electron affinity will override any electron affinity specification for Material2. This has an impact on any calculation where this materials electron affinity is used and must be considered when specifying Schottky barriers contacted to this materials. See Example 4 for more details on Schottky barrier considerations

---

Note: Remember, if the ALIGN parameter is not specified on the MATERIAL statement, BLAZE will use the Affinity Rule and either the default electron affinity or the affinity assigned using the AFFINITY parameter on the MATERIAL statement to calculate the conduction band offsets.

EXAMPLE 2:

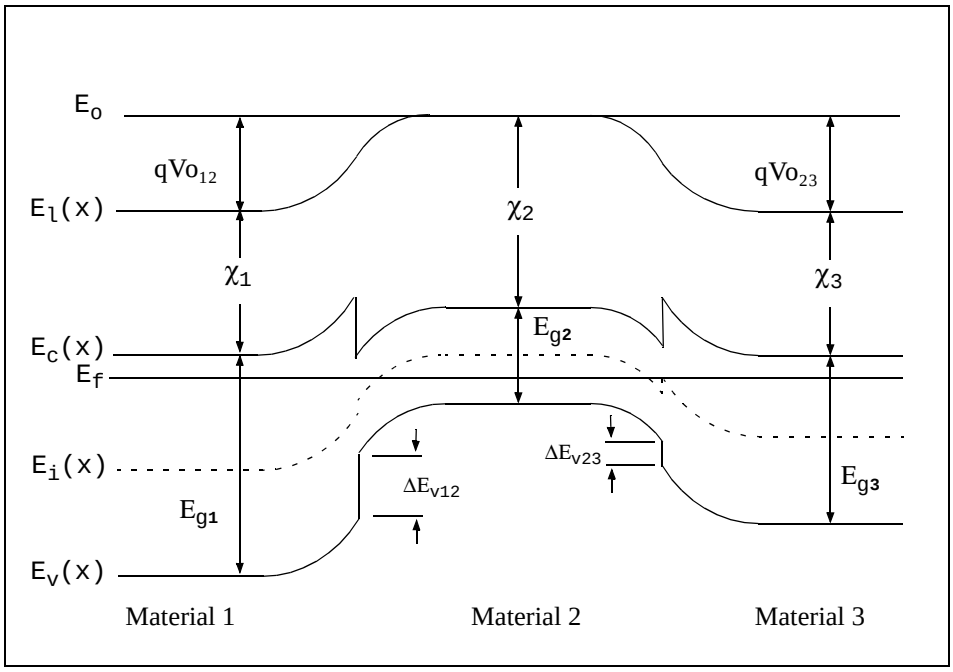

Figure 5-3: Band diagram of three material system (lowest E_g in center)

Figure 5-3 details a heterostructure device consisting of three semiconductors with different bandgaps E_{g1} , E_{g2} and E_{g3} and electron affinities χ_1, χ_2 and χ_3. This is similiar to the band diagram of a Double Heterojunction Bipolar Transistor. For this example, $E_{g1} > E_{g2} < E_{g3}$ and $\chi_1 < \chi_2 > \chi_3$.

Allocating the conduction band offsets using the affinity rule:

$$\Delta E_{c12} = \chi_2 - \chi_1 \qquad 5\text{-}4$$

and

$$\Delta E_{v12} = \Delta E_{g12} - \Delta E_{c12} \qquad 5\text{-}5$$

for the heterojunction between Material1 and Material2 and

$$\Delta E_{c23} = \chi_2 - \chi_3 \qquad 5\text{-}6$$

and

$$\Delta E_{v23} = \Delta E_{g23} - \Delta E_{c23} \qquad 5\text{-}7$$

for the heterojunction between Material 2 and Material 3.

Using the ALIGN parameter on the MATERIAL statement:

Notice that the reference material, the material with the smallest bandgap in this case Material2, is located between the two larger bandgap materials, Material 1 and Material 3.

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset for this heterojunction. Defining the ALIGN parameter on the MATERIAL statement for Material 1 using

```
MATERIAL NAME=Material1 ALIGN=0.8
```

then

$$\Delta E_{c12} = (E_{g1} - E_{g2}) \cdot 0.80 \quad \text{5-8}$$

Internally, the affinity of Material 1 is adjusted so that ΔE_{c12} equals this value.

Let's assign 70% of the bandgap difference between Material 3 and Material 2 to the conduction band offset for this heterojunction. Defining the ALIGN parameter on the MATERIAL statement for Material 3 using

```
MATERIAL NAME=Material3 ALIGN=0.70
```

then

$$\Delta E_{c23} = (E_{g3} - E_{g2}) \cdot 0.70 \quad \text{5-9}$$

Internally, the affinity of Material 3 is adjusted so that ΔE_{c23} equals this value.

These new values of electron affinity for Material1 and Material3 will override any electron affinity specification for these materials. This has an impact on any calculation where these materials electron affinity is used and must be considered when specifying Schottky barriers contacted to these materials. See Example 4 for more details on Schottky barrier considerations.

Manually Adjusting Material Affinity

Assigning the conduction band offsets for each heterojunction is accomplished by setting the electron affinities for Material 1 and Material 3 using the AFFINITY parameter on the MATERIAL statement. The electron affinity for Material 1 and Material 3 are adjusted relative to Material 2 by the amount of the desired conduction band offset for each heterojunction. Since Material 2 affinity is larger than that for Material 1 and Material 3, the affinities for Material 1 and Material 3 are reduced (relative to Material 2) to provide the desired conduction band offset.

Let's assume an electron affinity for Material 2 of 4eV (~ that of GaAs). Let's decide that between Material 1 and Material 2, the conduction band offset is 0.3eV and that between Material 3 and Material 2, the conduction band offset is 0.2eV. Then, for Material 1

```
MATERIAL NAME=Material1 AFFINITY=3.7
```

and for Material 3

```
MATERIAL NAME=Material3 AFFINITY=3.8
```

This is the easiest method to define the conduction band offsets for multiple materials. This value of electron affinity will override any electron affinity specification. This has an impact on any calculation where this materials electron affinity is used and must be considered when specifying Schottky barriers contacted to this materials. See Example 4 for more details on Schottky barrier considerations.

Note: The band offsets are always defined with reference to the conduction band. Therefore, if a specific valence band offset is required, the appropriate conduction band offset should be calculated from the desired valence band offset and the materials bandgap.

EXAMPLE 3:

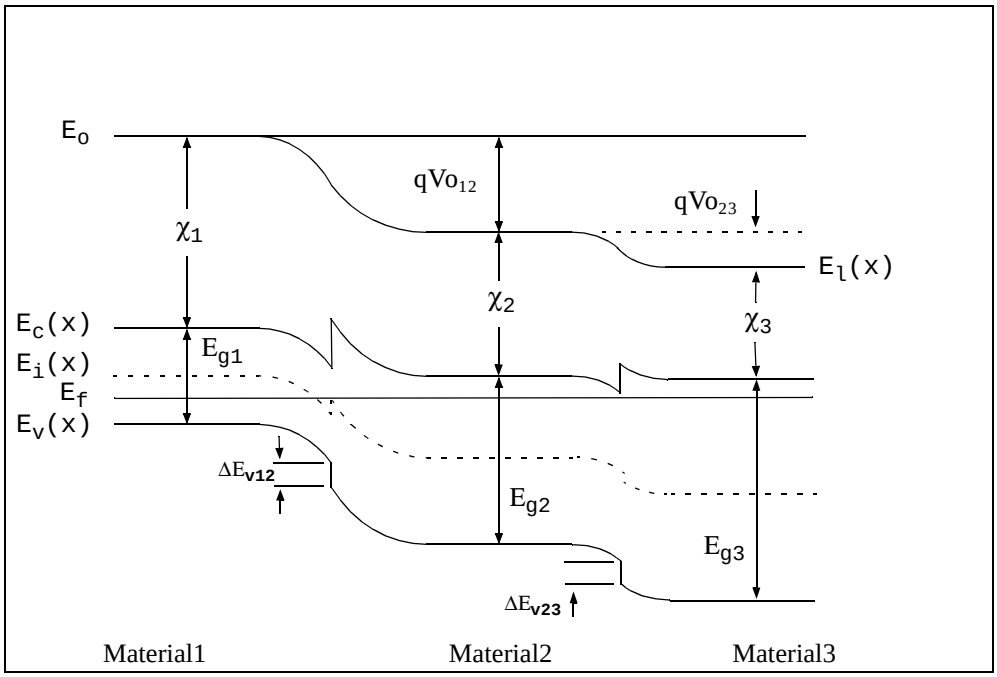

Figure 5-4: Band diagram of three material system (lowest E_g not in center)

Figure 5-4 details a heterostructure device consisting of three semiconductors with different bandgaps E_{g1} , E_{g2} and E_{g3} and electron affinities χ_1, χ_2 and χ_3. This is similiar to the Example 2 except that the narrow bandgap material is not located in between the other larger bandgap materials. As will be seen, this adds extra complexity to the conduction and valence band offset calculations. For this example, $E_{g1} < E_{g2} < E_{g3}$ and $\chi_3 < \chi_2 < \chi_1$.

Allocating the conduction band offsets using the affinity rule,

$$\Delta E_{c12} = \chi_1 - \chi_2 \qquad 5\text{-}10$$

and

$$\Delta E_{v12} = \Delta E_{g12} - \Delta E_{c12} \qquad 5\text{-}11$$

for the heterojunction between Material1 and Material2 and

$$\Delta E_{c23} = \chi_2 - \chi_3 \qquad 5\text{-}12$$

and

$$\Delta E_{v23} = \Delta E_{g23} - \Delta E_{c23} \quad \text{5-13}$$

for the heterojunction between Material2 and Material3.

Using the ALIGN parameter on the MATERIAL statement

Notice that the reference material, the material with the smallest bandgap in this case Material2, is not shared between the two larger bandgap materials, Material1 and Material3. This will be important in calculating the conduction band offsets for the heterojunction formed by Material3 and Material 2 (the one in which the reference material is not present).

Let's assign 80% of the bandgap difference between Material 1 and Material 2 to the conduction band offset for this heterojunction. Since the reference material is one of the materials of this heterojunction, we can procede as before. Defining the ALIGN parameter on the MATERIAL statement for Material 2 using

```
MATERIAL NAME=Material2 ALIGN=0.8
```

then

$$\Delta E_{c12} = (E_{g2} - E_{g1}) \cdot 0.80 \quad \text{5-14}$$

Internally, the affinity of Material 2 is adjusted so that ΔE_{c12} equals this value.

Let's assign 70% of the bandgap difference between Material3 and Material2 to the conduction band offset for this heterojunction. Since the reference material is not one of the materials in this heterojunction, another procedure will be used. Since BLAZE always uses the bandgap of the reference material (the smallest bandgap material in overall structure) when calculating the conduction band offset using the ALIGN parameter on the MATERIAL statement, the actual value for the ALIGN parameter needs to be calculated as follows:

$$\text{ALIGN} = (\Delta E_{g32} / \Delta E_{g31}) \cdot \text{FRACTION} \quad \text{5-15}$$

where FRACTION is the desired fraction of the bandgap difference between Material3 and Material2 that will appear in the conduction band. ΔE_{g23} is the bandgap difference for the actual heterojunction and ΔE_{g31} is the bandgap difference using the reference material. Once calculated, this value for the ALIGN parameter can be used on the MATERIAL statement for Material3. FRACTION and ALIGN will only be equal when the reference material is one of the two materials in the heterojunction. For this example, lets assume that

$$\Delta E_{g32} = 0.2 \quad \text{5-16}$$

and

$$\Delta E_{g31} = 0.4 \quad \text{5-17}$$

then for a desired conduction band offset fraction of 0.70,

$$\text{ALIGN} = (\Delta E_{g32} / \Delta E_{g31}) \cdot \text{FRACTION} = (0.2/0.4) \times 0.70 = 0.35 \quad \text{5-18}$$

The proper value of the ALIGN parameter reflecting the desired conduction band offset can be assigned as

```
MATERIAL NAME=Material3 ALIGN=0.35
```

This assigns 70% of the bandgap difference between Material3 and Material2 as the conduction band offset. Internally, the affinity of Material 3 is adjusted so that ΔE_{C23} equals this value.

Note: Calculating ALIGN in this manner is only necessary when the reference material is not in contact with the material for which the ALIGN parameter will be specified.

These new values of electron affinity for Material2 (from the first heterojunction band offset calculation) and Material 3 (from the second heterojunction band offset calculation) will override any electron affinity specification for these materials. This has an impact on any calculation where these materials electron affinity is used and must be considered when specifying Schottky barriers contacted to these materials. See Example 4 for more details on Schottky barrier considerations.

Manually Adjusting Material Affinity

Assigning the conduction band offsets for each heterojunction is accomplished by setting the electron affinities for Material2 and Material3 using the AFFINITY parameter on the MATERIAL statement. The electron affinity for Material2 is adjusted relative to Material1 and Material3 is adjusted relative to Material2 by the amount of the desired conduction band offset for each heterojunction. Since Material1 affinity is larger than that for Material2 and Material2 affinity is larger than that for Material3, the affinities for Material2 and Material3 are reduced to provide the desired conduction band offsets.

Let's assume an electron affinity for Material 1 of 4eV (~ that of GaAs). Let's decide that between Material1 and Material2, the conduction band offset is 0.3eV and that between Material 2 and Material 3, the conduction band offset is 0.2eV. Then, for Material 2

```
MATERIAL NAME=Material2 AFFINITY=3.7
```

and for Material 3

```
MATERIAL NAME=Material3 AFFINITY=3.5
```

This is the easiest method to define the conduction band offsets for multiple materials. This value of electron affinity will override any electron affinity specification. This has an impact on any calculation where this materials electron affinity is used and must be considered when specifying Schottky barriers contacted to this materials. See Example 4 for more details on Schottky barrier considerations.

Note: The band offsets are always defined with reference to the conduction band. Therefore, if a specific valence band offset is required, the appropriate conduction band offset should be calculated from the desired valence band offset and the materials bandgap.

EXAMPLE 4:

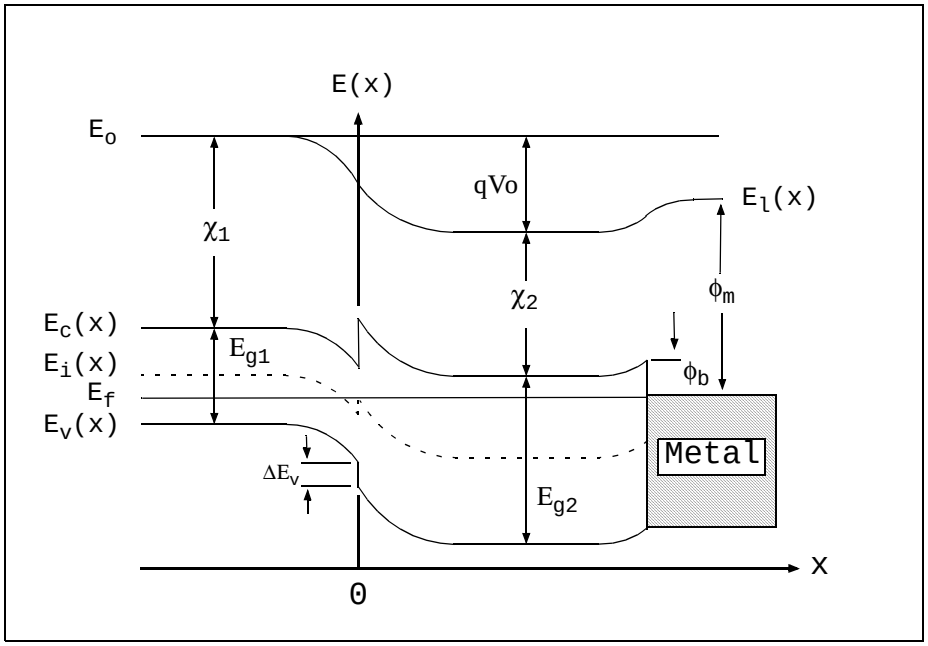

Figure 5-5:

Figure 5-5 details a heterostructure device consisting of two semiconductors with different bandgaps E_{g1} and E_{g2} and electron affinities $\chi_1$and χ_2 and a Schottky barrier. For this example, $E_{g1} < E_{g2}$ and $\chi_2 < \chi_1$.

This example will first define the heterojunction band offsets and then the Schottky barrier height. Schottky contact barrier heights are calculated by BLAZE using the metal work function and the semiconductor electron affinity as

$$\phi_b = \phi_m - \chi_s \qquad 5\text{-}19$$

where ϕ_b is the Schottky barrier height, ϕ_m is the work function of the metal, and χ_s is the semiconductor electron affinity. ϕ_m is set using using the WORKFUN parameter on the CONTACT statement. Therefore, the semiconductor electron affinity as modified or defined during the heterojunction alignment process plays an important role in determining the value of the metal workfunction needed to provide the desired barrier height. Let's assume for this example that a Schottky barrier height of 0.2eV is desired and calculate the appropriate metal workfunction for each case.

Using the affinity rule for the heterojunction

$$\Delta E_c = \chi_1 - \chi_2 \qquad 5\text{-}20$$

and

$$\Delta E_v = \Delta E_g - \Delta E_c \qquad 5\text{-}21$$

ΔE_c is the amount of the conduction band discontinuity at the heterointerface and ΔE_v is is the amount of the valence band discontinuity.

Note: Remember the Affinity Rule is invoked for a material as long as the ALIGN parameter is NOT specified on the MATERIAL statement for that material.

Let's use an electron affinity for Material1 of 4eV and for Material2 of 3.5eV. Since the affinity of the material on which the Schottky barrier is formed was not modified with this method of alignment, the metal work function needed to provide for a Schottky barrier height of 0.2eV is

$$\phi_m = 3.5 + 0.2 = 3.7 \quad \text{5-22}$$

This value can now be assigned to the WORKFUN parameter on the CONTACT statement as

```
CONTACT NUMBER=1 WORKFUN=3.7
```

This produces a Schottky barrier height of 0.2eV between the metal and Material 2.

Using the ALIGN parameter on the MATERIAL statement:

Let's assign 80% of the bandgap difference between Material1 and Material2 to the conduction band offset, an electron affinity for Material 1 of 4eV, and ΔE_g of 0.2eV. Defining the ALIGN parameter on the MATERIAL statement for Material 2 using

```
MATERIAL NAME=Material2 ALIGN=0.80
```

Then

$$\Delta E_c = (E_{g2} - E_{g1}) \cdot 0.80 = 0.2 \cdot 0.80 = 0.16 \quad \text{5-23}$$

Internally, the affinity of Material2 is reduced by 0.16eV so

$$\chi_2 = \chi_1 - \Delta E_c = (4 - 0.16) = 3.84 \quad \text{5-24}$$

This value of electron affinity can be used to assign the proper value of WORKFUN on the CONTACT statement to provide for a Schottky barrier height of 0.2eV.

$$\phi_m = 3.84 + 0.2 = 4.04 \quad \text{5-25}$$

This value can now be assigned to the WORKFUN parameter on the CONTACT statement as

```
CONTACT NUMBER=1 WORKFUN=4.04
```

producing a Schottky barrier height of 0.2eV between the metal and Material 2.

Manually Adjusting Material Affinity

Let's assign a conduction band offset between Material 1 and Material 2 of 0.15eV, an electron affinity for Material 1 of 4eV, and the desired Schottky barrier height of 0.2eV. The affinity for Material 2 is calculated from the affinity of Material 1 and the desired conduction band offset as

$$\chi_2 = \chi_1 - 0.15 = 4 - 0.15 = 3.85 \quad \text{5-26}$$

This is then used to assign the value of AFFINITY using,

```
MATERIAL NAME=Material2 AFFINITY=3.85
```

This value of electron affinity can now be used to calculate the metal workfunction necessary to produce a Schottky barrier height of 0.2eV as

$$\phi_m = 3.85 + 0.2 = 4.05 \quad \text{5-27}$$

This value can now be assigned to the WORKFUN parameter on the CONTACT statement as

```
CONTACT NUMBER=1 WORKFUN=4.05
```

This produces a Schottky barrier height of 0.2eV between the metal and Material2.

The Drift Diffusion Transport Model

Drift-Diffusion with Position Dependent Band Structure

The current continuity equations for electrons and holes, and the Poisson equation (see Chapter 3) are the same as for the homogeneous case. (Although the changing dielectric constant is taken into account). However, the current density expressions must be modified to take into account the nonuniform band structure. This procedure starts with the current density expressions:

$$\vec{J}_n = -\mu_n n \nabla \phi_n \quad \text{5-28}$$

$$\vec{J}_p = -\mu_p n \nabla \phi_p \quad \text{5-29}$$

where ϕ_n and ϕ_p are quasi-Fermi potentials.

$$\phi_n = \frac{1}{q} E_{FN} \quad \text{5-30}$$

$$\phi_p = \frac{1}{q} E_{FP} \quad \text{5-31}$$

The conduction and valence band edge energies can be written as:

$$E_c = q(\psi_0 - \psi) - \chi \quad \text{5-32}$$

$$E_v = q(\psi_0 - \psi) - \chi - E_g \quad \text{5-33}$$

where:

ψ_0 is some reference potential.

χ is the position-dependent electron affinity.

Eg is the position-dependent bandgap.

ψ_0 can be selected in the form:

$$\psi_0 = \frac{\chi_r}{q} + \frac{kT_L}{q}\ln\frac{N_{cr}}{n_{ir}} = \frac{\chi_r + E_g}{q} - \frac{kT_L}{q}\ln\frac{N_{vr}}{n_{ir}} \quad 5\text{-}34$$

where n_{ir} is the intrinsic carrier concentration of the arbitrarily selected reference material, and r is the index that indicates that all of the parameters are taken from reference material. Fermi energies are expressed in the form:

$$E_{FN} = E_c + kT_L \ln\frac{n}{N_c} - kT_L \ln\gamma n \quad 5\text{-}35$$

$$E_{FP} = E_v + kT_L \ln\frac{n}{N_v} - kT_L \ln\gamma n \quad 5\text{-}36$$

The final terms in Equations 5-35 and 5-36 are due to the influence of Fermi-Dirac statistics. These final terms are defined as follows:

$$\gamma_n = \frac{F_{1/2}(\eta_n)}{e^{\eta_n}},\ \eta_n = \frac{E_{FN} - E_c}{kT_L} = F_{1/2}^{-1}\left(\frac{n}{N_c}\right) \quad 5\text{-}37$$

$$\gamma_p = \frac{F_{1/2}(\eta_p)}{e^{\eta_p}},\ \eta_p = \frac{E_v - E_{FP}}{kT_L} = F_{1/2}^{-1}\left(\frac{p}{N_v}\right) \quad 5\text{-}38$$

where:

N_c and N_v are position-dependent.

γ_n=γ_p= 1 for Boltzmann statistics.

By combining Equations 5-28 to 5-38, the following expressions for current densities can be obtained.

$$\vec{J}_n = kT_L\mu_n\nabla n - q\mu_n n\nabla\left(\psi + \frac{kT_L}{q}\ln\gamma_n + \frac{\chi}{q} + \frac{kT_L}{q}\ln\frac{N_c}{n_{ir}}\right) \quad 5\text{-}39$$

$$\vec{J}_p = kT_L\mu_p\nabla n - q\mu_p p\nabla\left(\psi + \frac{kT_L}{q}\ln\gamma_p + \frac{\chi + E_g}{q} + \frac{kT_L}{q}\ln\frac{N_v}{n_{ir}}\right) \quad 5\text{-}40$$

The Thermionic Emission Transport Model

It is possible to activate an alternative current density expression (Wu and Yang [72]) , which takes into account thermionic emission dominated current in abrupt heterojunctions. This equation applies only at the node points along the interface of the heterojunction and takes the form;

$$\vec{J}_n = \frac{q(kT_L)^2}{2\pi^2 h^3}\eta m^* \left(\frac{n^+}{\gamma_n^+ N_C^+} - \frac{n^-}{\gamma^- N_C^-} \exp\left(\frac{\chi^+ - \chi^-}{kT_L} \right) \right) \quad 5\text{-}41$$

where J_n is the current density from the " -" region to the "+" region.

The "+" and "-" regions are defined such that χ^+ is less than χ^-. The variable η is a coefficient which accounts for quantum mechanical reflection and tranmission at the interface, and:

$$m^* = \frac{2}{\frac{1}{m^+} + \frac{1}{m^-}} \quad 5\text{-}42$$

where:

m^+ is the effective electron mass in the "+" region.

m^- is the effective electron mass in the "-" region.

Expression 5-41 can be rewritten in the following form:

$$\vec{J}_n = A^{**} \cdot T_L^2 \left(\frac{n^+}{\gamma_n^+ N_C^+} - \frac{n^-}{\gamma^- N_C^-} \exp\left(\frac{\chi^+ - \chi^-}{kT_L} \right) \right) \quad 5\text{-}43$$

where:

$$A^{**} = \frac{2}{\frac{1}{A^{*+}} + \frac{1}{A^{*-}}} \quad 5\text{-}44$$

and $A^* = A\eta$ is the effective Richardson constant, which can be specified from input language in the `MATERIAL` statement. An analagous expression is used for holes.

The Physical Models

In Chapter 3 a comprehensive set of physical models has already been described for silicon. However, it is known that for III-V, II-VI and terniary compounds that special consideration are required such as for mole fraction dependence, material properties, etc. The following sections will describe the *material dependent* physical models that have been implemented into BLAZE to account for these effects.

First a description of the common mobility model equations is given which are applicable to all materials unless otherwise stated in following material sections. Following this are sections describing the physical models on a material per material basis. For each material there are descriptions of the models for bandgap narrowing, electron affinity, density of states, dielectric permittivity and low field mobility.

Common Physical Models

Low Field Mobility Models

The default low field mobility models used for most materials in BLAZE are given by the following:

$$\mu_{n0}(T_L) = \mathbf{MUN}\left(\frac{T_L}{300}\right)^{\mathbf{TMUN}} \quad 5\text{-}45$$

$$\mu_{p0}(T_L) = \mathbf{MUP}\left(\frac{T_L}{300}\right)^{\mathbf{TMUP}} \quad 5\text{-}46$$

Where T_L is the temperature in degrees Kelvin and the parameters MUN, MUP, TMUN and TMUP are user-definable as shown in Table 5-1.

Note: All the mobility models described in Chapter 3 may also be used within BLAZE except for the TASCH model. However, all default coefficients exist only for Silicon and therefore are not suitable for compound materials.

Table 5-1. User Specifiable Parameters for Equations 5-45 and 5-46

Statement	Parameter	Units
MOBILITY	MUN	cm2/V·s
MOBILITY	TMUN	
MOBILITY	MUP	cm2/V·s
MOBILITY	TMUP	

Parallel Electric Field-Dependent Mobility

Two types of electric field-dependent mobility models are used in ATLAS/BLAZE. These models are a standard mobility model and a negative differential mobility model. Both of these models contain appropriate default values of parameters for different materials. The user must specify what type of mobility will be used for each material and what material parameters they wish to alter.

The standard mobility model that takes account of velocity saturation is defined according to;

$$\mu_n(E) = \mu_{n0}\left[\frac{1}{1+\left(\frac{\mu_{n0}E}{\mathbf{VSATN}}\right)^{\mathbf{BETAN}}}\right]^{1/\mathbf{BETAN}} \quad 5\text{-}47$$

$$\mu(E) = \mu_{p0}\left[\frac{1}{1+\left(\frac{\mu_{p0}E}{\mathbf{VSATP}}\right)^{\mathbf{BETAP}}}\right]^{1/\mathbf{BETAP}} \quad 5\text{-}48$$

where VSATN and VSATP are the saturation velocities for electrons and holes, BETAN and BETAP are constants given in Table 5-2 and $\mu_{n0,p0}$ are the electron and hole low field mobilities. This model is activated by the parameter FLDMOB or EVSATMOD=0 of the MODEL statement.

Table 5-2. User Specifiable Parameters for Equation 5-47 and 5-48

Statement	Parameter	Units
MOBILITY	BETAN	
MOBILITY	BETAP	
MOBILITY	VSATN	cm/s
MOBILITY	VSATP	cm/s

The negative differential mobility model of Barnes et. al., [24] has been implemented to account for certain devices where the carrier drift velocity peaks at some electric field before reducing as the electric field increases. This model takes account of this through the carrier mobility with equations of the form:

$$\mu_n(E) = \frac{\mu_{n0} + \frac{\mathbf{VSATN}}{E}\left(\frac{E}{\mathbf{ECRITN}}\right)^{\mathbf{GAMMAN}}}{1+\left(\frac{E}{\mathbf{ECRITN}}\right)^{\mathbf{GAMMAN}}} \quad 5\text{-}49$$

$$\mu_p(E) = \frac{\mu_{p0} + \frac{\mathbf{VSATP}}{E}\left(\frac{E}{\mathbf{ECRITP}}\right)^{\mathbf{GAMMAP}}}{1+\left(\frac{E}{\mathbf{ECRITP}}\right)^{\mathbf{GAMMAP}}} \quad 5\text{-}50$$

where VSATN and VSATP are the electron and hole saturation velocities, E_0 is a constant, and $\mu_{n0,p0}$ are the low-field electron and hole mobilities.

Table 5-3. User Specifiable Parameters for Equation 5-49 and 5-50

Statement	Parameter	Default	Units
MOBILITY	ECRITN	$4.0*10^3$	V/cm
MOBILITY	ECRITP	$4.0*10^3$	V/cm
MOBILITY	GAMMAN	4.0	
MOBILITY	GAMMAP	1.0	
MOBILITY	VSATN	Eq. 5-20	cm/s
MOBILITY	VSATP	Eq. 5-20	cm/s

Note: The negative differential mobility model introduces an instability in the solution process and is not recommended for general use. Only activate this model in cases where the device operation directly depends on negative differential mobility (e.g. a Gunn diode).

For both the standard and negative differential models an empirical temperature-dependent model for saturation velocity in GaAs [13] is implemented according to;

$$\mathbf{VSATN} = \mathbf{VSATP} = 11.3\times10^{6} - 1.2\times10^{4}\ T_L \qquad 5\text{-}51$$

where VSATN and VSATP are expressed in cm/sec and T_L is the temperature in degrees Kelvin.

Alternatively, the saturation velocities can be set to constant values using the VSATN and VSATP parameters of the MATERIAL statement.

Velocity Saturation with Energy Balance

When the energy balance transport model is activated the mobility can be made a function of carrier energy. In Chapter 3 physical models for the dependence of carrier mobility on carrier energy were introduced. The same models are applicable for use within BLAZE with one additional model which applies when the negative differential moibility model is used.

The carrier temperature dependence is activated when the paramter EVSATMOD=1 on the MODEL statement. This model can be derived in a similar fashion as in the case of EVSATMOD=0 described in Chapter 3. These expressions however require several piecewise approximations which are too complex to reproduce in this manual. Sufice it to say that these piece-wise expressions provide a continuous velocity saturation model for mobility versus carrier temperature completely analogous to the expressions for drift diffusion given in Equations 3-202 **and** 3-203 of Chapter Three.

Recombination and Generation Models

The recombination and generation models for compound semiconductors are the same as the models previously described in Chapter 3. Default parameters for different materials are automatically used, unless new values are specified by the user. The default parameter values are listed in Appendix B.

Material Dependent Physical Models

GaAs

Bandgap Narrowing

Following Lundstrom [71], the bandgap narrowing effects are important only for p-type regions. By default BLAZE uses the bandgap narrowing values shown in Table 5-4.

Table 5-4. Default Bandgap Narrowing Values

Concentration cm^{-3}	Bandgap Narrowing meV
$1.0*10^{18}$	31.0
$2.0*10^{18}$	36.0
$4.0*10^{18}$	44.2
$6.0*10^{18}$	48.5
$8.0*10^{18}$	51.7
$1.0*10^{19}$	54.3
$2.0*10^{19}$	61.1
$4.0*10^{19}$	64.4
$6.0*10^{19}$	61.9
$8.0*10^{19}$	56.9
$1.0*10^{20}$	53.2
$2.0*10^{20}$	18.0

Low Field Mobilty

The mobility in GaAs can be made concentration dependent by setting the CONMOB parameter of the MODEL statement.

In this model mobility is interpolated from the values in Table 5-5.

Table 5-5. Default Concentration Dependent Mobility for GaAs

Concentration(cm-3)	Mobility in GaAs (cm^2/v-s)	
	Electrons	Holes
$1.0*10^{14}$	8000.0	390.0
$2.0*10^{14}$	7718.0	380.0
$4.0*10^{14}$	7445.0	375.0
$6.0*10^{14}$	7290.0	360.0
$8.0*10^{14}$	7182.0	350.0
$1.0*10^{15}$	7300.0	340.0
$2.0*10^{15}$	6847.0	335.0
$4.0*10^{15}$	6422.0	320.0
$6.0*10^{15}$	6185.0	315.0
$8.0*10^{15}$	6023.0	305.0
$1.0*10^{16}$	5900.0	302.0
$2.0*10^{16}$	5474.0	300.0
$4.0*10^{16}$	5079.0	285.0
$6.0*10^{16}$	4861.0	270.0
$8.0*10^{16}$	4712.0	245.0
$1.0*10^{17}$	4600.0	240.0
$2.0*10^{17}$	3874.0	210.0
$4.0*10^{17}$	3263.0	205.0
$6.0*10^{17}$	2950.0	200.0
$8.0*10^{17}$	2747.0	186.9
1.0*1018	2600.0	170.0
$2.0*10^{18}$	2060.0	130.0

Table 5-5. Default Concentration Dependent Mobility for GaAs		
Concentration(cm-3)	Mobility in GaAs (cm^2/v-s)	
	Electrons	Holes
$4.0*10^{18}$	1632.0	90.0
$6.0*10^{18}$	1424.0	74.5
$8.0*10^{18}$	1293.0	66.6
$1.0*10^{19}$	1200.0	61.0

If MODEL ANALYTIC is specified, the program will use:

$$\mu_{n,\,p} = 940 + \frac{8000 + 940}{1 + \left(\frac{Nt}{2.8*10^{16}}\right)^{0.75}} \quad \text{5-52}$$

where Nt is the total impurity concentration.

III-V and II-VI Materials

Al(x)Ga(1-x)As System

The Al(x)Ga(1-x)As material system is commonly used for the fabrication of heterojunction devices. These materials are available in BLAZE by specifying the material name GaAs, AlGaAs, or AlAs. As a ternary material system, different material properties are obtained by adjusting the molar fraction of Aluminum and Gallium. This mole fraction is represented by the x as written in Al(x)Ga(1-x)As. GaAs material parameters are identical to the those of AlGaAs with mole the fraction x set equal to zero. AlAs material parameters are identical to the those of AlGaAs with mole the fraction x set equal to one. Fundamental in the proper simulation with the AlGaAs material system is the relationship between this mole fraction x, and the material parameters for that composition. In the following sections, the relationship between mole fraction and material parameters for the AlGaAs material system will be described.

Note: Users should not use this material system to form GaAs by setting x=0 but instead specify GaAs as the material.

Bandgap

There are three primary conduction bands in the AlGaAs system that depending on mole fraction, determine the bandgap. These are named Gamma, L, and X. The default bandgaps for each of these conduction band valleys are as follows:

$$E_{g\Gamma} = EG300 + x.composition \cdot (0.574 + 0.055 \cdot x.composition) \quad \text{5-53}$$

$$E_{gL} = 1.734 + x.composition \cdot (0.574 + 0.055 \cdot x.composition) \quad \text{5-54}$$

$$E_{gX} = 1.911 + x.composition \cdot (0.005 + 0.245 \cdot x.composition) \quad 5\text{-}55$$

The bandgap used for any given Al concentration is the minimum as calculated from these equations. **EG300** is the bandgap at 300K and specified on the material statement. **x.composition** is the Aluminum mole fraction and may be user defined on the REGION statement.

The temperature dependence of the bandgap is calculated iaccording to;

$$E_g(T_L) = E_g(300) + EGALPHA \cdot \left[\frac{300^2}{300 + EGBETA} - \frac{T_L^2}{T_L + EGBETA}\right] \quad 5\text{-}56$$

The value of Eg(300) is taken as the minimum of $E_{g\Gamma}$, E_{gX}, and E_{gL}. The default temperature dependent bandgap parameters for AlGaAs are listed in Table 5-1

Statement	Parameter	Default	Units
MATERIAL	EG300	1.59	eV
MATERIAL	EGALPHA	$5.405*10^{-4}$	eV/K
MATERIAL	EGBETA	204	K

Electron Affinity

As indicated in the introduction, the semiconductor electron affinity χ is a key parameter for determining the alignment of heterojunctions. For AlGaAs, χ is a function of $E_{g\Gamma}$ and is given by:

$$\chi_{AlGaAs} = 4.07 - 0.85 \cdot (E_{g\Gamma}(x.composition) - E_{gGaAs}) \quad 5\text{-}57$$

Density of States and Effective Mass

The valence and conduction band densities of states, N_C and N_V, are calculated from the effective masses according to the following equations;

$$N_c = \left(\frac{2\pi m_e^* kT_L}{h^2}\right)^{\frac{3}{2}} \quad 5\text{-}58$$

$$N_v = \left(\frac{2\pi m_h^* kT_L}{h^2}\right)^{\frac{3}{2}} \quad 5\text{-}59$$

For the AlGaAs system the conduction band and valence band effective masses, for electrons and holes, are given by:

$$m_e^* = 0.067 \quad 5\text{-}60$$

$$m_h^* = (m_{lh}^{1.5} + m_{hh}^{1.5})^{\frac{2}{3}} \quad 5\text{-}61$$

where m_{lh} and m_{hh} represent the light hole and heavy hole effective masses which for this system are;

$$m_{lh} = 0.082 \quad 5\text{-}62$$

$$m_{hh} = 0.45 \quad 5\text{-}63$$

Dielectric Permittivity

The default static dielectric constant for AlGaAs is given by:

$$\varepsilon_{AlGaAs} = 13.8 + 2.9 \cdot x.composition \quad 5\text{-}64$$

Low Field Mobility

The default low field electron mobility for AlGaAs is a function of the composition fraction x within the system. The following equations outline this relationship;

AlGaAs Low Field Mobility	Mole Fraction Range
$\mu_n = 8000 - (1.818 \times 10^4 \cdot x.composition)$	$(0 < x.composition < 0.429)$
$\mu_n = 90 + 1.1435 \times 10^4 \cdot (x.composition - 0.46)^2$	$(0.429 < x.composition < 0.46)$
$\mu_n = 90 + 3.75 \times 10^4 \cdot (x.composition - 0.46)^2$	$(0.46 < x.composition < 0.5)$
$\mu_n = 200 - (2 / (x.composition - 0.46))$	$(0.5 < x.composition < 1.0)$

In(x)Ga(1-x)As(y)P(1-y) System

The In(x)Ga(1-x)As(y)P(1-y) material system is commonly used for the fabrication of heterojunction devices. These include laser diodes, photodiodes, Gunn diodes, and high speed heterostructure transistors. As a quaternary material, two different mole fraction parameters, x and y, are necessary to specify any particular combination. This produces a wide array of InGaAsP materials and characteristics. Of particular interest in this system are materials that are lattice matched to InP.

The default material characteristics in BLAZE for the InGaAsP system correspond to composition fractions x and y that yield InGaAsP material that is lattice matched to InP. The relationship between x and y that satisfy this condition is given by:

$$x = \frac{0.1896 \cdot y.composition}{0.4176 - (0.0125 \cdot y.composition)} \qquad 0 < y.composition < 1 \qquad 5\text{-}65$$

Many of the parameter models for the In(x)Ga(1-x)As(y)P(1-y) system are functions of the composition fraction y.composition only. The composition fraction x.composition can be deduced from the preceeding relationship. Again, the default material characteristics in BLAZE for the InGaAsP system correspond to composition fractions x and y that yield InGaAsP material that is lattice matched to InP.

Note: Users should not use this material system to form GaAs by setting x=0 and y=1 but instead specify GaAs as the material.

Bandgap

The default energy bandgap for the InP lattice matched In(x)Ga(1-x)As(y)P(1-y) system used in BLAZE is given by:

$$E_g(InGaAsP) = 1.35 + x.composition \cdot (0.642 + (0.758 \cdot x.composition)) \qquad 5\text{-}66$$

$$+ (0.101 \cdot y.composition - 1.101) \cdot y.composition$$

$$- (0.28 \cdot x.composition - 0.109 \cdot y.composition + 0.159) \cdot x.composition \cdot y.composition$$

Electron Affinity

The electron affinities for materials in the InP lattice matched InGaAsP system are derived from conduction band offsets and from the assumption that the affnity of InP is 4.4eV. The default conduction band edge offset between lattice matched InGaAsP and InP is then

$$\Delta E_c = 0.268 \cdot y.composition + 0.003 \cdot (y.composition)^2 \qquad 5\text{-}67$$

Density of States and Effective Mass

The density of states is defined, as before, as a function of the effective masses of electrons and holes according to Equation 3-29. For the InGaAsP system the default conduction and valence band effective masses, for electrons and holes, are given by the following.

For the conduction band;

$$m_e^* = 0.08 - (0.116 \cdot y.composition) + (0.026 \cdot x.composition) \qquad 5\text{-}68$$

$$- 0.059 \cdot (x.composition \cdot y.composition)$$

$$+ (0.064 + 0.02 \cdot x.composition) \cdot (y.composition)^2$$

For the valence band the hole effective mass is defined by;

$$m_h^* = (m_{lh}^{1.5} + m_{hh}^{1.5})^{\frac{2}{3}} \quad 5\text{-}69$$

where the default light hole effective mass is given by

$$m_{lh} = 0.120 - (0.116 \cdot y.composition) + 0.03 \cdot (x.composition)^2 \quad 5\text{-}70$$

and the default heavy hole effective mass is a constant and is given by;

$$m_{lh} = 0.46 \quad 5\text{-}71$$

Dielectric Permittivity

The default static dielectric constant for lattice matched InGaAsP to InP is given by

$$\begin{aligned}\varepsilon_{InGaAsP} = {} & (14.6 \cdot (1 - x.composition) \cdot y.composition) \\ & + 12.5 \cdot (1 - x.composition) \cdot (1 - y.composition) \\ & + 13.18 \cdot x.composition \cdot y.composition \\ & + 11.11 \cdot x.composition \cdot (1 - y.composition)\end{aligned} \quad 5\text{-}72$$

Low Field Mobility

The default low field mobility parameters for electrons and holes for lattice matched InGaAs are given by linear interpolations from the binary compunds GaAs and InP. The following formulas are used:

$$\mu_{n1} = 33000 + (8500 - 33000) \cdot x.composition \quad 5\text{-}73$$

$$\mu_{p1} = 460 + (400 - 460) \cdot x.composition \quad 5\text{-}74$$

$$\mu_{n2} = 4600 + (300 - 4600) \cdot x.composition \quad 5\text{-}75$$

$$\mu_{p2} = 150 + (100 - 150) \cdot x.composition \quad 5\text{-}76$$

$$\mu_{n0} = \mu_{n1} + (y.composition - 1)(\mu_{n2} - \mu_{n1}) \quad 5\text{-}77$$

$$\mu_{p0} = \mu_{p1} + (y.composition - 1)(\mu_{p2} - \mu_{p1}) \quad 5\text{-}78$$

The Si(1-x)Ge(x) System

Advances in the growth of Silicon and Si(1-x)Ge(x) alloys have allowed the potential for using bandgap engineering to construct heterojunction devices such as HBTs and HEMTs using these materials.

BLAZE supports the SiGe material system by providing composition dependent material parameters. These parameters are accessed by specifying the material name SiGe. The following sections describe the functional relationship between Ge mole fraction x, and the SiGe material characteristics necessary for device simulation.

Bandgap

Bandgap is one of the most fundamental parameters for any material. For SiGe, the dependence of the bandgap on the Ge mole fraction, **x.composition,** is divided into ranges as follows:

$$E_g = 1.08 + x.composition \cdot ((0.945 - 1.08)/0.245) \quad 5\text{-}79$$

for $x \leq 0.245$

$$E_g = 0.945 + (x.composition - 0.245) \cdot (0.87 - 0.945)/(0.35 - 0.245) \quad 5\text{-}80$$

or $0.245 < x \leq 0.35$

$$E_g = 0.87 + (x.composition - 0.35) \cdot (0.78 - 0.87)/(0.5 - 0.35) \quad 5\text{-}81$$

for $0.35 < x \leq 0.5$

$$E_g = 0.78 + (x.composition - 0.5) \cdot (0.72 - 0.78)/(0.6 - 0.5) \quad 5\text{-}82$$

for $0.5 < x \leq 0.6$

$$E_g = 0.72 + (x.composition - 0.6) \cdot (0.69 - 0.72)/(0.675 - 0.6) \quad 5\text{-}83$$

for $0.6 < x \leq 0.675$

$$E_g = 0.69 + (x.composition - 0.675) \cdot (0.67 - 0.69)/(0.735 - 0.675) \quad ; \quad 5\text{-}84$$

for $0.675 < x \leq 0.735$

$$E_g = 0.67 \quad ; \quad 5\text{-}85$$

for $0.735 < x \leq 1$

The temperature dependence of the bandgap of SiGe is calculated the same as for Silicon using except that EGALPHA and EGBETA are a function of Ge mole fraction x as follows;

$$E_g(T_L) = E_g + \mathbf{EGALPHA}\left[\frac{300^2}{300 + \mathbf{EGBETA}} - \frac{T_L^2}{T_L + \mathbf{EGBETA}}\right] \quad 5\text{-}86$$

$$\mathbf{EGALPHA} = (4.73 + x.composition \cdot (4.77 - (4.73)) \times 10^4) \quad 5\text{-}87$$

$$\mathbf{EGBETA} = 626 + x.composition \cdot (235 - 636) \quad 5\text{-}88$$

where E_g is dependent upon the mole fraction as above.

Electron Affinity

The electron affinity χ of SiGe is taken to be constant with respect to composition.

Density of States

The density of states for SiGe is defined differently compared to the previous materials by not being a function of the effective masses. Instead the density of states have been made to depend upon the Ge mole fraction **x.composition** according to;

$$N_c = 2.8\times10^{19} + \text{x.composition} \cdot (1.04\times10^{19} - 2.8\times10^{19}) \quad \text{5-89}$$

$$N_v = 1.04\times10^{19} + \text{x.composition} \cdot (6.0\times10^{19} - 1.04\times10^{19}) \quad \text{5-90}$$

Dielectric Function

The compositional dependence of the static dielectric constant of SiGe is given by

$$\varepsilon = 11.8 + 4.2 \cdot \text{x.composition} \quad \text{5-91}$$

Low Field Mobility

No specific SiGe low field mobility models have been implemented into BLAZE.

Velocity Saturation

In SiGe the temperature dependent velocity saturation, used in the field dependent mobility model,is defined by the following equations;

$$\textbf{VSATN} = 1.38*10^{7} \cdot \sqrt{\left(\tanh\left(\frac{175}{T_L}\right)\right)} \quad \text{5-92}$$

$$\textbf{VSATP} = 9.05*10^{6} \cdot \sqrt{\left(\tanh\left(\frac{312}{T_L}\right)\right)} \quad \text{5-93}$$

Note: All other defaults used for SiGe are taken from Silicon

Silicon Carbide (SiC)

Recently there has been a great deal of interest in silicon carbide materials for high power, high temperature applications. The main characteristics of silicon carbide are that they have a very wide bandgap, high thermal conductivity, high saturation velocity, and high breakdown strength. Silicon carbide is commercially available in two polytypes called 6H-SiC and 4H-SiC. The ATLAS framework supports both of these polytypes which are termed aSiC and bSiC for 6H-SiC and 4H-SiC respectively. The following paragraphs describe the material defaults for these materials.

Band Parameters for SiC

SiC band parameter equations are identical to those used for Silicon but with the values adjusted for 4H and 6H-SiC. The physical band parameter values are summarised in Tables B-18 and B-19 of Appendix B.

SiC Mobility Parameters

Isotropic Mobiliy

By default mobility is assumed to be entirely isotropic in nature. that is there is no directional component. The default low field mobilities of electrons and holes for 4H and 6H SiC are shown in Table 5-6 and 5-9 below.

Table 5-6. 6H-SiC Low Field Mobility

Statement	Parameter	Default	Units
MOBILITY	MUN	330	cm2/V·s
MOBILITY	TMUN		
MOBILITY	MUP	60	cm2/V·s
MOBILITY	TMUP		

Table 5-7. 4H-SiC Low Field Mobility

Statement	Parameter	Default	Units
MOBILITY	MUN	1000	cm2/V·s
MOBILITY	TMUN		
MOBILITY	MUP	50	cm2/V·s
MOBILITY	TMUP		

Anisotropic Mobility

The mobility behaviour within SiC is now known to be anisotropic in nature which dramatically alters the electrical performance of a device. An isotropic model has been implemented into ATLAS to correctly model this behaviour. Following the ideas of Lindefelt [136]and Lades [137] the mobility within the drift diffusion equations has been made a tensor property. As a result the mobility has become;

$$\mu = \begin{bmatrix} \mu_1 & 0 & 0 \\ 0 & \mu_1 & 0 \\ 0 & 0 & \mu_2 \end{bmatrix} \quad \text{5-94}$$

where μ_1 represents the mobility defined in one plane and μ_2 the mobility defined in a second plane. In the case of SiC μ_1 represents the mobility of plane <1100> whilst μ_2 represents the mobility of plane <1000>. These mobilities are defined for both holes and electrons.

Defining Anisotropic Mobility in ATLAS

To define a material with anisotropic mobility it is necessary to specify two MOBILITY statements. In each statement the parameters N.ANGLE and P.ANGLE are used to specify the direction in which that particular mobility is to apply. The following is an example of how this is done;

```
# FIRST DEFINE MOBILITY IN PLANE <1100>
#
MOBILITY MATERIAL=B-SIC VSATN=2E7 VSATP=2E7 BETAN=2 BETAP=2 \
   MU1N.CAUG=10 MU2N.CAUG=410 NCRITN.CAUG=13E17 \
   DELTAN.CAUG=0.6 GAMMAN.CAUG=0.0 \
   ALPHAN.CAUG=-3 BETAN.CAUG=-3 \
   MU1P.CAUG=20 MU2P.CAUG=95 NCRITP.CAUG=1E19 \
   DELTAP.CAUG=0.5 GAMMAP.CAUG=0.0 \
   ALPHAP.CAUG=-3 BETAP.CAUG=-3
#
# NOW DEFINE MOBILITY IN PLANE <1000>
#
MOBILITY MATERIAL=B-SIC N.ANGLE=90.0 VSATN=2E7 VSATP=2E7 BETAN=2 BETAP=2 \
   MU1N.CAUG=5 MU2N.CAUG=80 NCRITN.CAUG=13E17 \
   DELTAN.CAUG=0.6 GAMMAN.CAUG=0.0 \
   ALPHAN.CAUG=-3 BETAN.CAUG=-3 \
   MU1P.CAUG=2.5 MU2P.CAUG=20 NCRITP.CAUG=1E19 \
   DELTAP.CAUG=0.5 GAMMAP.CAUG=0.0 \
   ALPHAP.CAUG=-3 BETAP.CAUG=-3
```

Impact Ionization and Thermal Parameters

The equations governing these effects are identical to those for Silicon but with adjusted coefficients. Please refer to Appendix B for a list of all these parameters.

Simulating Heterojunction Devices with Blaze

Defining Material Regions with Positionaly Dependent Band Structure

Step Junctions

The easiest way to define a device with positionally dependent band structure is to specify two adjacent semiconductor regions with dissimilar bandgap. In this case BLAZE would simulate an abrupt heterojunction between the two materials. As an example, suppose the user wanted to simulate an abrupt heterojunction parallel to the x-axis at a location of y=0.1 microns. For values of y greater than 0.1 the user might specify, for example, GaAs. For values of y less than 0.1 the user might specify AlGaAs with a composition fraction of 0.3. The following statements would specify this situation:

```
REGION Y.MIN=0.1 MATERIAL=GaAs
REGION Y.MAX=0.1 MATERIAL=AlGaAs X.COMPOSITION=0.3
```

This fragment specifies that the two regions form an abrupt heterojunction at Y=0.1. The first region is composed of GaAs while the second is composed of AlGaAs.

These two material names are used by BLAZE to choose default material models for the two regions. A complete list of the materials which are available in ATLAS/BLAZE is given in Appendix B of this manual. For the AlGaAs region a composition fraction of 0.3 is specified.

Graded Junctions

A grading can be applied to this heterojunction with a simple modification. For example:

```
REGION Y.MIN=0.1 MATERIAL=GaAs
REGION Y.MAX=0.1 MATERIAL=AlGaAs \  X.COMPOSITION=0.3 GRAD.3=0.01
```

specifies that the composition fraction of the AlGaAs region decreases from 0.3 at y=0.1 microns to 0.0 at y=0.11 microns. The GRAD. parameter specifies the distance over which the mole fraction reduces to zero. The GRAD. parameter is indexed such that GRAD.1 corresponds to the Y.MIN side of the region, GRAD.2 corresponds to the X.MAX side of the region, GRAD.3 corresponds to the Y.MAX side of the region, and GRAD.4 corresponds to the X.MIN side of the region. In most cases the GRAD.n parameter acts to increase the size of the region. By default the GRAD.n parameters are set to zero and all heterojunctions are abrupt. It should also be noted that the GRAD. parameter acts just like the other region geometry parameters in that later defined regions overlapping the graded part of the region will overlap the grading. If in the previous example, the grading had been applied to the GaAs region it would be over lapped by the AlGaAs region. This would have produced an abrupt interface. A solution would be to limit Y.MAX in the AlGaAs region to 0.09. Care should always be taken to specify regions in the proper order to avoid such problems.

Along similar lines, the overlapping of regions can be used to an advantage in forming graded heterojunctions between two materials in the same system with different non-zero composition fractions. For example:

```
REGION Y.MIN=0.1 MATERIAL=AlGaAs X.COMPOSITION=0.3 GRAD.1=0.02
REGION Y.MAX=0.11 MATERIAL=AlGaAs X.COMPOSITION=0.1
```

specifies a graded heterojunction with a composition of 0.3 at y = 0.1 falling to 0.1 at y = 0.11.

By default heterojunctions in BLAZE are simulated using a drift-diffusion approximation. To simulate thermionic emission at an abrupt heterojunction, the EMISS. parameter must be set. The following illustrates specification of a thermionic heterojunction:

```
REGION Y.MIN=0.1 MATERIAL=GaAs
REGION Y.MAX=0.1 MATERIAL=AlGaAs X.COMPOSITION=0.3 EMISS.3
```

This is similar to the first example of the previous section. The only difference is that in the second REGION statement the EMISS.3 parameter is set. This envokes the thermionic emission model. The EMISS.n parameter is indexed in the same way as the GRAD.n paramter.

Defining Materials and Models

Materials

For example to set the bandgap for the material "InP" the following syntax should be used:

```
MATERIAL MATERIAL=InP EG300=1.35
```

Models

BLAZE has two ways of simulating the physical effects of variations in semiconductor composition. For relatively gradual variations in composition the standard modifications to the drift-diffusion equations can be considered adequate for simulation purposes. For abrupt heterojunctions it has been suggested that thermionic emission may be the dominant factor in the behavior of heterojunction behavior.

Individual material parameters and models can be defined for each material or region. These models are set in the MATERIAL, MODEL, and IMPACT statements.

This statement uses the MATERIAL parameter to select all regions composed of the material "InP". The bandgap in these regions will be set to 1.35. The parameters of a particular region can be set in two ways. The first is by way of the region index, as in:

```
MODEL REGION=1 BGN
```

In this case, the band gap narrowing model is enabled in the region indexed number 1.

The region name may also be used as in the following:

```
IMPACT NAME=substrate SELB
```

This example turns on the Selberherr impact ionization model in the region named "substrate". Finally, parameters can be set for all regions and materials by omitting the MATERIAL, REGION, or NAME parameters, as in the following:

```
MODEL BGN
```

This statement sets the bandgap narrowing model for all regions and materials.

Parser Functions

The use of parser functions requires knowledge of the C programming language, please consult Appendix A for a description of the parser functions. To specify completely arbitrary spatial variation of varying composition fraction, a parser function can be used. To define the parser function for composition fraction, write a C function describing the composition fraction as a function of position. A template for the function named "composition" is provided with this release of ATLAS. Once the function "composition" is defined, it should be stored in a file. In order to use the function for composition the parameter, F.COMPOSIT should be set to the file name of the function.

Chapter 6: GIGA

Overview

GIGA extends ATLAS to account for lattice heat flow and general thermal environments. GIGA implements Wachutka's thermodynamically rigorous model of lattice heating which accounts for Joule heating, heating and cooling due to carrier generation and recombination, and the Peltier and Thomson effects. GIGA accounts for the dependence of material and transport parameters on the lattice temperature, and supports the specification of general thermal environments using a combination of realistic heatsink structures, thermal impedances, and specified ambient temperatures. GIGA works with both S-PISCES and BLAZE, and with both the drift-diffusion and energy balance transport models.

Before continuing with this chapter you should be familiar with ATLAS, as described in Chapter 2, and with either S-PISCES (Chapter 4) or BLAZE (Chapter 5).

Applications

A major application of GIGA is the simulation of high-power structures including bipolar, MOS, IGBT, and thyristor devices. Another important application is the simulation of electrostatic discharge (ESD) protection devices. Thermal effects are also important in SOI device operation due to the low thermal conductivity of the buried oxide, and in devices fabricated in III-V material systems due to the relatively low thermal conductivity of these materials.

Recent studies have demonstrated that accounting self-consistently for lattice heating is necessary for accurate simulation of bipolar VLSI devices. This is due to the sensitive temperature dependence of the carrier injection process. Since bipolar devices are key components of CMOS technologies, and because many devices can be impacted by parasitic bipolar effects, the applications of GIGA are very general.

Numerics

GIGA supplies numerical techniques that provide efficient and robust solution of the complicated systems of equations that result when lattice heating is accounted for. These numerical techniques include fully-coupled and block iteration methods. When GIGA is used in conjunction with the energy balance equations the result is a "six equation solver" that defines the state-of-the-art for general purpose device simulation.

Physical Models

The Lattice Heat Flow Equation

GIGA adds the heat flow equation to the primary equations that are solved by ATLAS. The heat flow equation has the form:

$$C\frac{\partial T_L}{\partial t} = \nabla(\kappa \nabla T_L) + H \qquad 6\text{-}1$$

where:

C is the heat capacitance per unit volume.

κ is the thermal conductivity.

H is the heat generation

T_L is the local lattice temperature.

The heat capacitance can be expressed as $C = \rho C_p$, where C_p is the specific heat and ρ is the density of the material.

Specifying the parameter LAT.TEMP on the MODELS statement includes the lattice heat flow equation in ATLAS simulations.

GIGA supports different combinations of models. For example, if the parameters HCTE and LAT.TEMP are specified in the MODEL statement, and both particle continuity equations are solved, all six equations are solved. If HCTE.EL is specified instead of HCTE, only five equations are solved and the hole temperature T_p is set equal to lattice temperature T_L.

Specifying Heat Sink Layers For Thermal Solutions

Regions may be defined for inclusion only in thermal calculations. These regions will typically consist of layers associated with heat sinks. They are defined using the REGION statement. Even though in reality the heat sink materials are typically metal conductors it is more convienient to specify these layers with the material type INSULATOR. This is because as insulators the program will only solve heat flow and not attempt to solve current continuity in these layers. The region number is subsequently used as an identifier when thermal conductivities and heat capacities are assigned to these regions.

The following statements specify two layers of a heatsink for inclusion in the thermal calculation.

```
REGION NUM=5 Y.MIN=0.5 Y.MAX=2.0 INSULATOR
REGION NUM=6 Y.MIN=2.0 Y.MAX=3.0 INSULATOR
```

Specifying Thermal Conductivity

For steady state calculations, the thermal conductivity κ is the only parameter of Equation 6-1 that must be specified for each material region in the structure. Thermal conductivity varies as a function of the lattice temperature. The temperature dependence of κ is modeled in GIGA as:

$$\kappa(T) = \frac{1}{TC.A + TC.B\ T_L + TC.C\ T_L^2} \qquad 6\text{-}2$$

where TC.A, TC.B and *TC.C* are constants for a particular material.

There are reasonable default values of the parameters TC.A, TC.B and TC.C for common semiconductor device materials such as silicon and silicon dioxide. For silicon, the default values are

those given by Selberherr[3]. If the default values are not used for all materials, the thermal conductivity parameters must be explicitly specified for all materials in the structure. The values of TC.A, TC.B and TC.C are specified using the MATERIAL statement.

The following statements would be used to specify the temperature dependent thermal conductivity of the regions defined previously:

```
MATERIAL REGION=5 TC.A=<n> TC.B=<n> TC.C=<n>
MATERIAL REGION=6 TC.A=<n> TC.B=<n> TC.C=<n>
```

Table 6-1. User Specifiable Parameters for Equation 6-2

Statement	Parameter	Units
MATERIAL	TC.A	cmK/w
MATERIAL	TC.B	cm/w
MATERIAL	TC.C	cm/(wK)

Specifying Heat Capacity

For transient calculations, it is necessary to specify heat capacities for every region in the structure. These are also functions of the lattice temperature, and are modeled as:

$$C = \mathbf{HC.A} + \mathbf{HC.B}\,T + \mathbf{HC.C}\,T^2 + \frac{\mathbf{HC.D}}{T^2} \quad (J/cm^3) \qquad 6\text{-}3$$

Default values of HC.A, HC.B, HC.C and HC.D are provided for common materials. These values can be specified on the MATERIAL statement

The following statements would be used to specify the temperature dependent heat capacities of the regions defined previously.

```
MATERIAL REGION=5 HC.A=<n> HC.B=<n> HC.C=<n> HC.D=<n>
MATERIAL REGION=6 HC.A=<>> HC.B=<n> HC.C=<n> HC.D=<n>
```

Table 6-2. User Specifiable Parameters for Equation 6-3

Statement	Parameter	Units
MATERIAL	HC.A	J/cm3/K
MATERIAL	HC.B	$J/cm^3/K^2$
MATERIAL	HC.C	$J/cm^3/K^2$
MATERIAL	HC.D	JK/cm^3

Non-Isothermal Models

Effective Density Of States

When lattice heating is specifed with the drift-diffusion transport model the effective density of states for electrons and holes are modeled as functions of the local lattice temperature as defined by Equations 3-29 and 3-30.

When lattice heating is specifed with the energy balance model the effective densities of states are modeled as functions of the local carrier temperatures T_n and T_p, as defined by Equations 3-92 and 3-93.

Nonisothermal Current Densities

When GIGA is used, the electron and hole current densities are modified to account for spatially varying lattice temperatures:

$$\vec{J}_n = -q\mu_n n(\nabla\phi_n + P_n\nabla T_L) \quad 6\text{-}4$$

$$\vec{J}_p = -q\mu_p p(\nabla\phi_p + P_p\nabla T_L) \quad 6\text{-}5$$

where:

P_n and P_p are the absolute thermoelectric powers for electrons and holes.

P_n and P_p are modeled as follows:

$$P_n = \frac{k}{q}\left(\ln\frac{n}{N_v} - \left(\frac{5}{2} + \mathbf{KSN}\right)\right) \quad 6\text{-}6$$

$$P_p = \frac{k}{q}\left(\ln\frac{p}{N_v} - \left(\frac{5}{2} + \mathbf{KSP}\right)\right) \quad 6\text{-}7$$

The default values forare given in Table 6-3.

Table 6-3. User Specifiable Parameters for Equations 6-6 and 6-7

Statement	Parameter	Default	Units
MODELS	KSN	-1	None
MODELS	KSP	-1	None

Heat Generation

When carrier transport is handled in the drift-diffusion approximation the heat generation term H used in Equation 6-1 has the form [60]:

$$H = \frac{\left|J_n^2\right|}{q\mu_n n} + \frac{\left|J_p^2\right|}{q\mu_n \rho} - T_L(\vec{J_n}\nabla P_n) + T_L(\vec{J_p}\nabla P_p) \qquad 6\text{-}8$$

$$+ q(R-G)\left\{\left[T_L\left(\frac{\partial \phi_n}{\partial T}\right)_{n,\,p} - \phi_n\right] - \left[T_L\left(\frac{\partial \phi_p}{\partial T}\right)_{n,\,p} - \phi_p\right]\right\}$$

$$-T_L\left[\left(\frac{\partial \phi_n}{\partial T}\right)_{n,\,p} + P_n\right]\mathrm{div}\ J_n - T_L\left[\left(\frac{\partial \phi_p}{\partial T}\right)_{n,\,p} + P_\rho\right]\mathrm{div} J_n$$

In the steady-state case, the last two terms are equal to zero and Equation 6-8 simplifies to:

$$H = \left[\frac{\overrightarrow{\left|J_n^2\right|}}{q\mu n} + \frac{\overrightarrow{\left|J_p^2\right|}}{q\mu_\rho p}\right] + q(R-G)[\phi_p - \phi_n + T_L(P_\rho - P_n)] - T_L(\vec{J_n}\nabla P_n + \vec{J_\rho}\nabla P_\rho) \qquad 6\text{-}9$$

The components of Equation 6-9 can be identified as follows:

$\left[\frac{\overrightarrow{J_\rho^2}}{q\mu_n n} + \frac{\overrightarrow{J_\rho^2}}{q\mu_\rho p}\right]$ is Joule heat.

$q(R-G)[\phi_\rho - \phi_n + T_L(P_\rho - P_n)]$ is recombination and generation heating and cooling.

$-T(\vec{J_n}\nabla P_n + \vec{J_\rho}\nabla P_\rho)$ accounts for the Peltier and Thomson effects.

A simple and intuitive form of H that has been widely used in the past is:

$$H = (\vec{J_n} + \vec{J_\rho}) \cdot \vec{E} \qquad 6\text{-}10$$

GIGA can use either Equation 6-8 or 6-9 for steady-state calculations. By default, Equation 6-9 is used. Equation 6-8 is used if the parameter `HEAT.FULL` is specified in the `MODELS` statement. The Peltier and Thomson terms are turned off by specifying the `^HEAT.PETHOM` parameter of the `MODELS` statement.

If the general expression shown in Equation 6-8 is used for the non-stationary case, the derivatives $\left(\frac{\partial \phi_n}{\partial T_L}\right)_{n,\,p}$ and $\left(\frac{\partial \phi_p}{\partial T_L}\right)_{n,\,p}$ are evaluated for the case of an idealized nondegenerate semiconductor and complete ionization.

The heat generation term H is always set equal to 0 in isulators.

When carrier transport is modeled in the energy balance approximation the following expression is used for H:

$$H = W_n + W_\rho + E_g \quad U, \tag{6-11}$$

where U, W_n and W_p are defined by formulas 3-94 through 3-96.

Thermal Boundary Conditions

At least one thermal boundary condition must be specified when the lattice heat flow equation is solved. The thermal boundary conditions used have the general form:

$$\sigma\left(\vec{J}^{u}_{tot} \cdot \hat{s}\right) = \alpha(T_L - T_{ext}) \tag{6-12}$$

where σ is either 0 or 1, $\vec{J}^{u}_{tot}$ is the total energy flux and $\hat{s}$ is the unit external normal of the boundary.

The projection of the energy flux onto s is:

$$\left(\vec{J}^{u}_{tot} \cdot \hat{s}\right) = -\kappa\frac{\partial T_L}{\partial n} + (T_L P_n + \phi_n)\hat{J}_n \cdot \hat{s} + (T_L P_\rho + \phi_\rho)(\hat{J}_\rho \cdot \hat{s}) \tag{6-13}$$

When σ = 0, Equation 6-12 specifies a Dirichlet (fixed temperature) boundary condition:

$$T_L = \textbf{TEMPER}. \tag{6-14}$$

where TEMPER may be defined on the THERMCONTACT statement as shown in the next section. Dirichlet boundary condtions may be specified for an external boundary (which may coincide with an electrode), or for an electrode that is inside the device.

When σ = 1, Equation 6-12 takes the form:

$$(\vec{J}_n \cdot \hat{s}) = \frac{1}{R_{th}}(T_L - \textbf{TEMPER}) \tag{6-15}$$

where the thermal resistance R_{th} is given by;

$$R_{th} = \frac{1}{\textbf{ALPHA}}$$

and ALPHA is user definable on the THERMCONTACT statement.

Specifying Thermal Boundary Conditions

Setting thermal boundary conditions is similar to setting electrical boundary conditions. The THERMCONTACT statement is used to specify the position of the thermal contact and any optional properties of the contact. Thermal contacts may be placed at any position in the device (including sidewalls). Equation 6-15 is used if a value is specified for α, otherwise Equation 6-14 is used.

The following command specifies that thermal contact number 1 is located between x=0 µm and x=2 µm at y=0 µm, and that the temperature at the contact is 300K.

```
THERMCONTACT NUM=1 X.MIN=0 X.MAX=2 Y.MIN=0 Y.MAX=0 TEMP=300
```

A simpler statement may be used if the coordinates of a thermal contact coincide with the coordinates of an electrical contact. In this case it is permissible to specify the location of the thermal contact by referring to the electrode number of the electrical contact. For example, the statement

```
THERMCONTACT NUM=1 ELEC.NUM=3 TEMP=400
```

specifies that thermal contact number 1 is located in the same position as electrode number 3 and that the contact temperature is 400K

Table 6-4. User Specifiable Parameters for Equations 6-14 and 6-15.

Statement	Parameter	Units
THERMCONTACT	TEMPER	K
THERMCONTACT	ALPHA	$W/(cm^2K)$

Representing a thermal environment in terms of thermal impedances leads to efficient solutions. However, thermal impedance representations are typically only approximations. Detailed thermal modeling, e.g. of the effect of heatsink design changes, typically requires the use of detailed modeling of thermal regions with specified external ambient temperatures.

Note: It is not possible to alter the value of a thermal resistor within a sequence of SOLVE statements. Users should rerun the input file whenever a thermal resistor is changed.

Temperature Dependent Material Parameters

GIGA automatically uses the built-in temperature dependence of the physical models that are specified. When lattice heating is specifed temperature dependent values are evaluated locally at each point in the device. When lattice heating is not solved for the models provide only global temperature dependence, i.e. all points in the device are assumed to be at the specified temperature.

The non-isothermal energy balance model uses the same carrier temperature dependencies of the mobility and impact ionization models as in the pure energy balance case, but with coefficients that depend on the local lattice temperature. Impact ionization coefficients depend on lattice temperature. Almost all other models and coefficients depend on lattice temperature.

When lattice heating is used there is no point in specifying models that do not include temperature dependence. For mobilities, you should not specify CONMOB, instead you also specify ANALYTIC or ARORA. For impact ionization coefficients you should specify the SELBERHERR model.

GIGA can account for the temperature dependence of the minority carrier lifetimes for electrons and/or holes. The LT.TAUN (electrons) and LT.TAUP (holes) parameters of the MATERIAL statement are used to select this model. The model is turned on whenever the value of LT.TAUN or LT.TAUP is greater than 0 which is the default.

The temperature dependence of electron and hole lifetimes in the SRH recombination model has the form:

$$\tau_n = \mathbf{TAUN0}\left(\frac{T}{300}\right)^{\mathbf{LT.TAUN}} \qquad 6\text{-}16$$

$$\tau_p = \mathbf{TAUP0}\left(\frac{T}{300}\right)^{\mathbf{LT.TAUP}} \qquad 6\text{-}17$$

Table 6-5. User Specifiable Parameters for Equations 6-16 and 6-17.			
Statement	**Parameter**	**Default**	**Units**
MATERIAL	TAUN0	1e-7	s
MATERIAL	TAUP0	1e-7	s
MATERIAL	LT.TAUN	0	
MATERIAL	LT.TAUP	0	

See Equations 3-213 and 3-214 for information regarding concentration dependent lifetimes.

C-interpreter Defined Peltier Coefficients

A C-interpreter functions is available that can be used to define the Peltier coefficient KSN and KSP as a function of the electron and hole carrier temperatures T_n and T_p. This is defined using the syntax;

```
MODELS F.KSN=<filename> F.KSP=<filename>
```

where the <filename> parameter is an ascii file containing the C-interpreter function. See Appendix A for more information regarding the C-interpreter.

Applications of GIGA

Power Device Simulation Techniques

This section contains a series of techniques which you may find useful when simulating typical power device structures. Not all of the features described below are specific to GIGA and are common to the ATLAS framework.

Floating Guard Rings

No special syntax is needed for the simulation of un-contacted doping areas used in floating guard rings. The program is able to simulate guard ring breakdown with the standard impact ionization models. In some extreme cases, convergence may be slow due to poor initial guesses. If convergence is slow, both GUMMEL and NEWTON should be specified in the METHOD statement.

Floating Field Plates

You should use the ELECTRODE statement to specify the field plate regions as electrodes. If these plates do not contact any semiconductor then these electrodes can be set to float in the same manner as EEPROM floating gates. The following statement line specifies that the field plate region PLATE1 is a floating field plate.

```
CONTACT NAME=PLATE1 FLOATING
```

If the plates do contact the semiconductor this syntax must not be used. Instead current boundary conditions are used at the electrode with zero current. See the Getting Started Chapter for more about floating electrodes.

External Inductors

Inductors are commonly used in the external circuits of power devices. The CONTACT statement can be used to set an inductor on any electrode. The following statement sets an inductance on the drain electrode of 3 μH/μm.

```
CONTACT NAME=DRAIN L=3E-3
```

The next statement is used to specify a non-ideal inductor with a resistance of 100 ohms*micron.

```
CONTACT NAME=DRAIN L=3E-3 R=100
```

More Information

Many examples using GIGA have been installed on your distribution tape or CD. These include power device examples but also SOI and III-V technologies. More information about the use of GIGA can be found by reading the text associated with each example.

[This page intentionally left blank]

Chapter 7: TFT

Polycrystalline and Amorphous Semiconductor Models

Introduction

Before continuing with this chapter, you should be familiar with ATLAS. If not, read Chapter 2 before proceeding further.

TFT is an ATLAS based product that enables the simulation of disordered material systems. Since TFT contains no material models it is necessary to combine the use of TFT with either S-PISCES or BLAZE under the ATLAS framework.

TFT enables the user to define an energy distribution of defect states in the bandgap of semiconductor materials. This is necessary for the accurate treatment of the electrical properties of such materials as polysilicon and amorphous silicon.

The syntax used by TFT is part of the ATLAS syntax. There is no need for users to learn a completely new simulator to run TFT simulations.

Simulating TFT Devices

This section is intended to illustrate the basic building blocks for a thin-film transistor simulation.

To use TFT you should specify the addition of defect states into the bandgap of a previously defined crystalline material. Throughout this section the example of polysilicon is used. However, amorphous silicon and other disordered materials are handled in a similar manner.

Defining The Materials

Defining a simple polysilicon thin-film transistor structure can be done using the command syntax of ATLAS. The MESH, X.MESH and Y.MESH statements can be used to construct a mesh as described in the "Getting Started" Chapter of this manual.

When defining the material regions the following syntax may be used.

```
REGION Y.MIN=-0.05 Y.MAX=0   OXIDE
REGION Y.MIN=0     Y.MAX=0.2 SILICON
REGION Y.MIN=0.2   Y.MAX=2   OXIDE
```

Note that the region is defined as silicon. It is also possible to define the material as polysilicon. However, users should note that it is the defect distribution rather than this initial material definition that will determine the electrical characteristics.

Defining The Defect States

Disordered materials contain a large number of defect states within the band gap of the material. In order to accurately model devices made of polycrystalline or amorphous materials it is necessary to use a *continuous* density of states. The DEFECT statement is used to specify the density of defect states (DOS) as a combination of exponentially decaying band tail states and Gaussian distributions of mid-gap states[60].

Density of States Model

The total density of states (DOS), $g(E)$, is assumed to be composed of four bands: two tail bands (a donor-like valence band and an acceptor-like conduction band) and two deep level bands (one acceptor-like and the other donor-like) which are modeled using a Gaussian distribution.

$$g(E) = g_{TA}(E) + g_{TD}(E) + g_{GA}(E) + g_{GD}(E) \quad 7\text{-}1$$

where E is the trap energy, E_C is the conduction band energy, E_V is the valence band energy and the subscripts T, G,A, D stand for tail, Gaussian (deep level), acceptor and donor states respectively.

$$g_{TA}(E) = \mathrm{NTA}\ \exp\left[\frac{E - E_c}{\mathrm{WTA}}\right] \quad 7\text{-}2$$

$$g_{TD}(E) = \mathrm{NTD}\ \exp\left[\frac{E_v - E}{\mathrm{WTD}}\right] \quad 7\text{-}3$$

$$g_{GA}(E) = \mathrm{NGA}\ \exp\left[-\left[\frac{E - \mathrm{EGA}}{\boldsymbol{WGA}}\right]^2\right] \quad 7\text{-}4$$

$$g_{GD}(E) = \mathrm{NGD}\ \exp\left[-\left[\frac{E - \mathrm{EGD}}{\mathrm{WGD}}\right]^2\right] \quad 7\text{-}5$$

For an exponential tail distribution, the DOS is described by its conduction and valence band edge intercept densities (NTA and NTD), and by its characteristic decay energy (WTA and WTD).

For Gaussian distributions, the DOS is described by its total density of states (NGA and NGD), its characteristic decay energy (WGA and WGD), and its peak energy/peak distribution (EGA and EGD). The user specifiable parameters for the density of defect states are shown in Table 7-1.

Table 7-1. User Specifiable Parameters for Equations 7-2 to 7-5

Statement	Parameter	Default	Units
DEFECT	NTA	$1.12*10^{21}$	cm^{-3}
DEFECT	NTD	$4.0*10^{20}$	cm^{-3}
DEFECT	NGA	$5.0*10^{17}$	cm^{-3}
DEFECT	NGD	$1.5*10^{18}$	cm^{-3}
DEFECT	EGA	0.4	eV
DEFECT	EGD	0.4	eV
DEFECT	WTA	0.025	eV

Table 7-1. User Specifiable Parameters for Equations 7-2 to 7-5			
Statement	**Parameter**	**Default**	**Units**
DEFECT	WTD	0.05	eV
DEFECT	WGA	0.1	eV
DEFECT	WGD	0.1	eV

Trapped Carrier Density

The electron and hole concentrations due to trapped states (n_T and p_T respectively) are given by:

$$n_T = \int_{E_V}^{E_C} g_{TA}(E) \cdot f_{t_{TA}}(E, n, p)\, dE + \int_{E_V}^{E_C} g_{GA}(E) \cdot f_{t_{GA}}(E, n, p)\, dE \quad . \qquad 7\text{-}6$$

$$p_T = \int_{E_V}^{E_C} g_{TD}(E) \cdot f_{t_{TD}}(E, n, p)\, dE + \int_{E_V}^{E_C} g_{GD}(E) \cdot f_{t_{GD}}(E, n, p)\, dE \quad . \qquad 7\text{-}7$$

$f_{t_{TA}}(E, n, p)$ and $f_{t_{GA}}(E, n, p)$ are probabilities of occupation for the tail and Gaussian acceptor DOS, while $f_{t_{TD}}(E, n, p)$ and $f_{t_{GD}}(E, n, p)$ are the equivalent for the donors.

In the steady-state case, the probability of occupation of a trap level at energy E is given by:

$$f_t(E, n, p) = \frac{v_n \sigma_n n + e_p}{v_n \sigma_n n + v_p \sigma_n p + e_n + e_p} \qquad 7\text{-}8$$

where v_n is the electron thermal velocity, v_p is the hole thermal velocity, σ_n is the capture cross-section for electrons and σ_p is the capture cross-section for holes. The emission rates for the electrons and holes (e_n and e_p) are shown in equations 7-9 and 7-10:

$$e_n = v_n \sigma_n n_i \exp\left[\frac{E - E_i}{kT}\right] \qquad 7\text{-}9$$

$$e_p = v_p \sigma_p n_i \exp\left[\frac{E_i - E}{kT}\right] \qquad 7\text{-}10$$

n_i is the intrinsic carrier concentration.

A different value of σ_n can be specified for each acceptor and donor tail and deep level state. These are specified on the DEFECTS statement as shown in Table 7-2. The values of σ_n are: SIGTAE is the acceptor tail state, SIGTDE is the donor tail state, SIGGAE is the deep level Gaussian state and SIGGDE is the Gaussian donor state. Similar expressions are used for σ_p (SIGTAH, SIGTDH, SIGGAH and SIGGDH).

Table 7-2. User Specifiable Parameters for Equations 7-8 to 7-10			
Statement	**Parameter**	**Default**	**Units**
DEFECT	SIGTAE	$1.0*10^{-16}$	cm^2
DEFECT	SIGTDE	$1.0*10^{-14}$	cm^2
DEFECT	SIGGAE	$1.0*10^{-16}$	cm^2
DEFECT	SIGGDE	$1.0*10^{-14}$	cm^2
DEFECT	SIGTAH	$1.0*10^{-14}$	cm^2
DEFECT	SIGTDH	$1.0*10^{-16}$	cm^2
DEFECT	SIGGAH	$1.0*10^{-14}$	cm^2
DEFECT	SIGGDH	$1.0*10^{-16}$	cm^2

Steady-state Trap Recombination

The Schockley-Reed-Hall model is used to determine the recombination rate due to the defect states.

$$R(n,p) = \sum_{l} \int_{E_V}^{E_C} \frac{(v_{th}\sigma_{ln}\sigma_{lp} n\ p - e_{ln}e_{lp})}{v_{th}(\sigma_{ln} n + \sigma_{lp} p) + e_{ln} + e_{lp}} g_l(E)\ dE \qquad 7\text{-}11$$

where the summation runs over all the donor and acceptor bands (l stands for TA, TD, GA, GD).

Transient Traps

An additional differential rate equation is solved whenever a transient simulation is performed This takes into account the time required for electron and hole to be emitted or captured. The probability of occupation becomes linked to the time dependent emission of electrons through the rate equation (Equation 7-12) and is no longer given by Equation 7-8. The rate equation is given by

$$\frac{d}{dt}(C_T) = (R_n(t) - G_n(t)) - (R_p(t) - G_p(t)) \qquad 7\text{-}12$$

where C_T is the trap concentration, $R_n(t)$ and $R_p(t)$ are the recombination rates of electrons and holes respectively while $G_n(t)$ and $G_p(t)$ are the electron and hole generation rates. C_T is equal to n_T when evaluating the electron current continuity equation and equal to p_T for the hole continuity equation.

$$R_n - G_n = \sum_{j} \int_{E_V}^{E_C} v_n \sigma_{jn} n(1 - f_{t_j}) - f_{tj}\ v_n \sigma_{jn} n_i \exp\left(\frac{E - E_i}{kT_L}\right) g_j(E)\ dE \qquad 7\text{-}13$$

$$R_p - G_p = \sum_k \int_{E_V}^{E_C} v_p \sigma_{kp} p \, f_{t_k} - (1 - f_{t_k}) \, v_p \sigma_{kp} n_i \exp\left(\frac{E_i - E}{kT_L}\right) g_k(E) \, dE \quad 7\text{-}14$$

The summation in Equation 7-13 runs over all acceptor bands (j stands for (TA and GA) and the summation in Equation 7-14 runs over all donor bands (k stands for TD and GD).

A transient trap simulation using this model is more time consuming than using the static model but gives a much more accurate description of the device physics. It may sometimes be acceptable to perform transient calculations using the static trap distribution and assume that traps reach equilibrium instantaneously. Specifying FAST on the DEFECTS statement will neglect the trap rate equation from the simulation.

Trap-Assisted Tunneling

The trap-assisted tunneling models can be used to include the effects of electrons tunneling through the bandgap via defects. This model is enabled by specifying TRAP.TUNNEL on the MODELS statement. The capture cross-sections (σ_n and σ_p) enhanced by the field-effect terms, Γ_n and Γ_p (as given by Hurkx et al[134]) in the following manner.

$$\sigma_n' = \frac{\sigma_n}{1 + \Gamma_n} \quad 7\text{-}15$$

$$\sigma_p' = \frac{\sigma_p}{1 + \Gamma_p} \quad 7\text{-}16$$

σ'_n and σ'_p replace σ_n and σ_p in the previous equations (Equations 7-8 to 7-14). Γ_n and Γ_p are given by Equations 3-67 and 3-69 in Chapter 3.

Continuous Defects

If CONTINUOUS is specified on the DEFECTS statement, then the integral equations for the charge and recombination are evaluated using a numerical integral scheme. In this case the parameters NUMA and NUMD (DEFECTS statement) correspond to the number of acceptor and donor energy level intervals used in the integral.

Discrete Defects

If CONTINUOUS is not specified on the DEFECTS statement, the equation is modeled using discrete energy levels. The integrals terms in Equations 7-6, 7-7, 7-11, 7-13 and 7-14 are replaced by summations which run over the number of discrete energy levels (NUMA and NUMD). The acceptor and donor density of states terms are integrated separately. For example, the equation for the electron trap concentration (Equation 7-6) is replaced by:

$$n_T = \sum_{i=0}^{\text{NUMA}} \left(f_{t_{TA}}(E_i, n, p) \cdot \int_{-\infty}^{+\infty} g_{TA}(E) \, dE + f_{t_{GA}}(E_i, n, p) \cdot \int_{-\infty}^{+\infty} g_{GA}(E) \, dE \right) \quad 7\text{-}17$$

Table 7-3. Additional Parameters for the DEFECTS Statement			
Statement	**Parameter**	**Default**	**Units**
DEFECT	FAST	FALSE	
DEFECT	CONTINUOUS	FALSE	
DEFECT	NUMA	12	
DEFECT	NUMD	12	

Syntax for a typical defect states definition is given below:

```
DEFECTS NTA=1.12E21 NTD=4.E20 WTA=0.025 WTD=0.05 \
     NGA=5.E17 NGD=1.5E18 EGA=0.4 EGD=0.4 \
     GA=0.1 WGD=0.1 SIGTAE=1.E-16 \
     SIGTAH=1.E-14 SIGTDE=1.E-14 \
     SIGTDH=1.E-16 SIGGAE=1.E-16 SIGGAH=1.E-14 \
     SIGGDE=1.E-14 SIGGDH=1.E-16
```

Figure 7-1 shows how the syntax is used to define the peak density of states and distribution widths for the two tail states and two Gaussian distributions.

Plotting The Density Of States Versus Energy

The parameters DFILE and AFILE were used to allow the user to specify output file names for capture of defect densities as a function of energy for donor states and acceptor states, respectively. For example if the user wants to look at the donor and acceptor defect distributions, the following line could be specified:

```
DEFECTS DFILE=donors AFILE=acceptors
```

Then the files *"donors"* and *"acceptors"* could be loaded into TONYPLOT to look at the distributions of donor and acceptor defects as a function of energy.

Using the C-Interpreter to define DEFECTS

The C-Interpreter can be used to define the defect states in the bandgap. The F.TFTDON and F.TFTACC parameters of the DEFECT statement indicate the filenames containing the C functions. See Appendix A for more information on using the C-Interpreter.

```
DEFECTS F.TFTDON=mydefects.c F.TFTACC=mydefects.c
```

The file *mydefects.c* will contain C functions for donor and acceptor defect densities as a function of energy. These user defined defects are added to the existing defect distribution. If users wish to use only their own function they should set the gaussian and tail functions to zero. In the following example defect states are defined in the file *tft.lib*. These are added to a zero background set using the tail and gaussian state syntax. The resultant distribution of defects versus energy can be plotted in the files *don.dat* and *acc.dat*.

```
DEFECTS F.TFTDON=tft.lib F.TFTACC=tft.lib DFILE=don.dat AFILE=acc.dat \
     NTA=0 NTD=0 WTA=1.0 WTD=1.0 \
     NGA=0 NGD=0 EGA=0.6 EGD=0.6 WGA=1 WGD=1 \
```

```
SIGTAE=1.E-16 SIGTAH=1.E-14 SIGTDE=1.E-14 SIGTDH=1.E-16
SIGGAE=1.E-16 SIGGAH=1.E-14 SIGGDE=1.E-14 SIGGDH=1.E-16
```

Setting Mobility and Other Models

TFT uses adaptations of the standard models of S-PISCES or BLAZE. An example of how to select the models and material parameters for polysilicon is:

```
MATERIAL MUN=300 MUP=30
MODELS SRH
```

Note: Concentration dependent mobility models (CONMOB, ANALYTIC, ARORA, KLA) should **not** be used as this will overwrite the low field mobilities set in the MATERIAL statement.

Typical mobility values for amorphous silicon can be set by:

```
MATERIAL MUN=20 MUP=1.5
```

Other models are also available in TFT. These include impact ionization and tunneling. These can be set by:

```
MODELS BBT.STD IMPACT
```

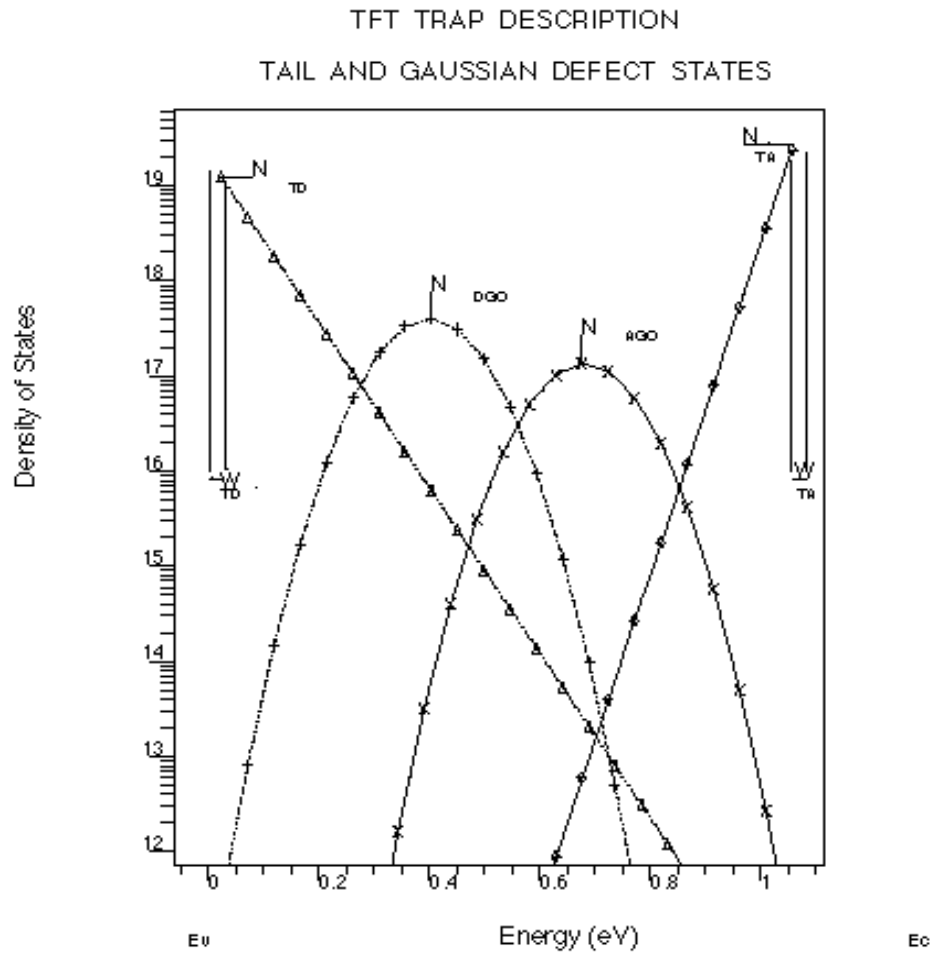

Figure 7-1: Syntax Guide to Define Two Tail States and Two Gaussian Distributions. NGA and NDG are the integrated values of the Gaussian distributions. Gaussians are entered on energies EGA and EGD respectively.

Chapter 8: Luminous

Introduction

Before continuing to the sections that follow, you should be familiar with ATLAS and either S-PISCES or BLAZE. If not, read Chapter 2 and either Chapter 4 or Chapter 5 of this manual before proceeding with this chapter.

LUMINOUS is a general purpose ray trace and light absorption program integrated into the ATLAS framework to run with device simulation products. When used with the S-PISCES or BLAZE, device simulators, LUMINOUS calculates optical intensity profiles within the semiconductor device, and converts these profiles into photogeneration rates in the device simulators. This unique coupling of tools allows the user to simulate electronic responses to optical signals for a broad range of optical detectors. These devices include but are not limited to: pn and pin photodiodes, avalanche photodiodes, Schottky photodetectors, MSMs, photoconductors, optical FETs, optical transistors, solar cells, and CCDs. The following sections address various types of optoelectronic devices. You can proceed to the sections most relevant to your application, but we strongly recommend you read the other sections as well. There is little overlap in the sections.

Simulation Method

Optoelectronic device simulation is split into two distinct models that are calculated simultaneously at each DC bias point or transient timestep:

1. Optical ray trace using real component of refractive index to calculate the optical intensity at each grid point
2. Absorption or photogeneration model using the imaginary component of refractive index to calculate a new carrier concentration at each grid point.

This is followed by an electrical simulation using SPISCES or BLAZE to calculate terminal currents.

Ray Tracing

Defining The Incident Beam

An optical beam is modeled as a collimated source using the BEAM statement. The origin of the beam is defined by parameters X.ORIGIN and Y.ORIGIN (see Figure 8-1). The ANGLE parameter specifies the direction of propagation of the beam relative to the x-axis. ANGLE=90 is vertical illumination from the top. MIN.WINDOW/MAX.WINDOW parameters specify the illumination window. As shown in Figure 8-1, the illumination window is "clipped" against the device domain so that none of the beam bypasses the device).

The beam is automatically split into a series of rays such that the sum of the rays covers the entire width of the illumination window. When the beam is split, ATLAS automatically resolves discontinuities along the region boundaries of the device.

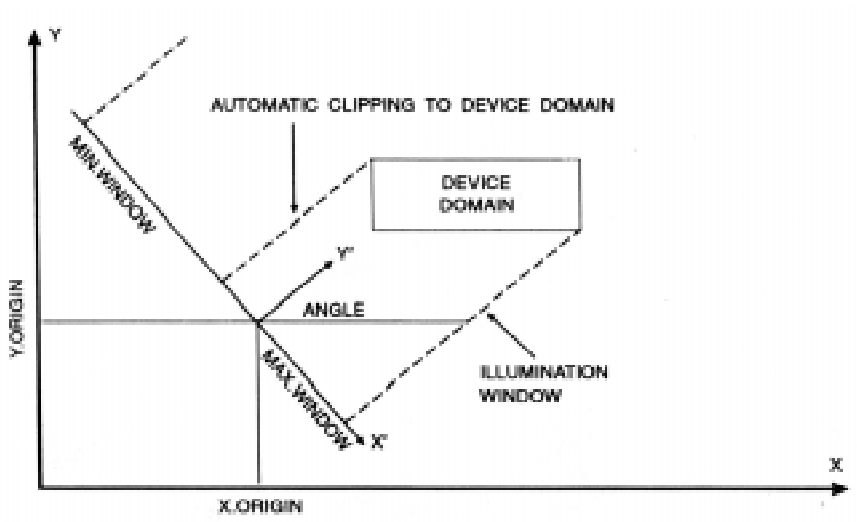

Figure 8-1: Optical Beam Geometry

Although the automatic algorithm is usually sufficient the user may also split the beam up into a number of rays using the RAYS parameter. Each ray will have the same width at the beam origin and the sum of the rays will cover the illumination window. Even when the RAYS parameter is specified, ATLAS will automatically split the rays in order to resolve the device geometry.

Ray Splitting At Interfaces

Rays are also split at interfaces between regions into a transmitted ray and a reflected ray. Figure 8-2 illustrates the difference between rays that are split to resolve the geometry and transmitted/reflected rays split at a region interfaces.

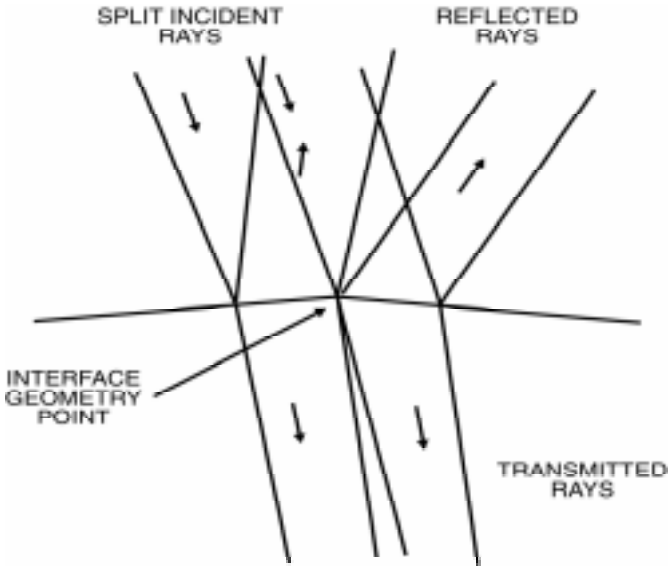

Figure 8-2: Reflected and Transmitted Rays

In Figure 8-2 the incident rays come in from the top left. They intersect an interface between two material regions with differing refractive indices. Within this interface lies a geometric point where

the normal to the interface changes. This implies that the angles of reflection and transmission will be different for light incident to the left of the point from light incident on the right. Thus, the incident rays are split to resolve the interface point. The second level of splitting occurs at the interface itself. Here the incident rays are split into reflected and transmitted rays.

Reflection and Transmission

Figure 8-3 shows the relationship between the angles of incidence (θ_i), reflection (θ_r), and transmission (θ_t) at the interface between two media. These coefficients are calculated as a function of the refractive indices in the two media.

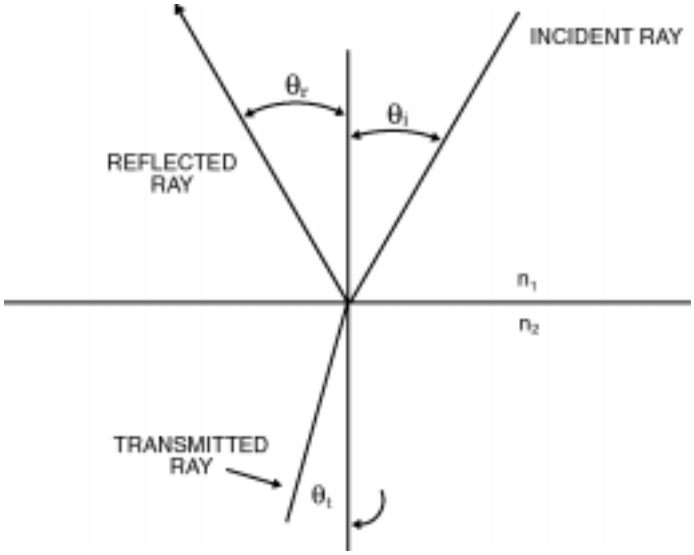

Figure 8-3: Angles of incidence, reflection and transmission

The reflection and transmission coefficients of the light for parallel and perpendicular polarization are calculated as in Equations 8-1 to 8-6.

$$E_r = \frac{n_1 \cos\theta_t - n_2 \cos\theta_i}{n_1 \cos\theta_t + n_2 \cos\theta_i} E_i \quad \text{(parallel polarization)} \tag{8-1}$$

$$E_t = \frac{2n_1 \cos\theta_i}{n_1 \cos\theta_t + n_2 \cos\theta_i} E_i \quad \text{(parallel polarization)} \tag{8-2}$$

$$E_r = \frac{n_1 \cos\theta_i - n_2 \cos\theta_t}{n_1 \cos\theta_i + n_2 \cos\theta_t} E_i \quad \text{(perpendicular polarization)} \tag{8-3}$$

$$E_t = \frac{2n_1 \cos\theta_i}{n_1 \cos\theta_i + n_2 \cos\theta_t} E_i \quad \text{(perpendicular polarization)} \qquad 8\text{-}4$$

$$R = \left(\frac{E_r}{E_i}\right)^2 \qquad 8\text{-}5$$

$$T = \left(\frac{E_t}{E_i}\right) \frac{n_2}{n_1} \qquad 8\text{-}6$$

where E_i is the incident intensity, E_r is the reflected intensity, E_t is the transmitted intensity. R is the reflection coefficient, T is the transmission coefficient, n_1 is the refractive index on the incident side and n_2 is the refractive index on the transmission side.

The angles of reflection and transmission are given in Equations 8-7 and 8-8:

$$\theta_r = \theta_i \qquad 8\text{-}7$$

$$n_1 \sin\theta_i = n_2 \sin\theta_t \qquad 8\text{-}8$$

where θ_i is the angle of incidence, θ_t is the angle of transmission and θ_r is the angle of reflection.

Specifying Reflections

By default no reflections are considered during the ray trace. The parameter REFLECTS=<i> is used to set an integer number of reflections to consider. Users should note that setting a very large number of reflections can lead to extremely long simulation times for the ray trace.

One very convienient way to overcome the long CPU times is to use the parameter MIN.POWER. This terminates each ray when the optical power falls to the fraction ofthe original power defined by this parameter.

Front Reflection

By default, reflection and refraction at the first interface (the initial interface with the device) are ignored. The first reflection coefficient is zero and the transmission coefficient is one. The polarization and angle of the transmitted ray at the first interface is identical to the polarization and angle of the incident beam.

If the FRONT.REFL parameter of the BEAM statement is specified, the transmission coefficient is calculated using Equations 8-1 to 8-6. When the transmission coefficient is calculated, it is assumed that the material outside the device domain is a vacuum. The transmitted rays are attenuated by the transmission coefficient, but the reflected ray is not traced.

Back Reflection

By default, the reflection at the back of the device are ignored. No reflected ray is traced once the back of the device is reached. If the BACK.REFL parameter is specified, the backside reflection coefficient is calculated (again assuming a vacuum outside the device) and the back-side reflected ray is traced.

Sidewall Reflection

By default, the reflection from the sides of the device are ignored. No reflected ray is traced back into the structure. As above BACK.REFL is used to enable the sidewall reflections assuming a vacuum outside the device.

Discontinuous Regions

It is possible to simulate devices electrically where a single region is defined as two or more separated areas. However the ray tracing algorithm does not support such structures. If a structure has two separate areas with the same region number the recommended approach is to use DEVEDIT to renumber the regions, perhaps even creating a new region number for each area.

Note this limitation is only for two separated areas with the same region number and not for two regions with different region numbers of the same material. This latter case can be simulated.

Anti-Reflective Coatings

It is a popular strategy to place anti-relective (AR) coatings on light detecting devices to improve device quantum efficiency. Such coatings rely on coherence effects to reduce the reflection coefficient between the detecting device and the ambient (i.e., air or vaccuum) in the direction of the light source. Typically, these AR coatings are composed of one or more layers of insulating materials that are one quarter optical wavelength thick and optically transparent to the wavelength in question.

Coherence effects are not currently accounted for in LUMINOUS. Thus, it is not a good idea for the user to place a layer of material explicitly into the simulated device structure to simulate an AR coating. With the addition of this layer, LUMINOUS will not properly simulate the reflectivity of the layer. In addition, this will introduce many additional nodes into the mesh that will most likely have essentially no effect on the electrical performance of the device. Instead, there are two models that can be used to model the effects of the AR coating on the reflectivity of the device.

First, there is a simple model for single layer AR coatings assuming no absorption in the AR coating. This case is illustrated in the following figure. Here, n1 is the index of refraction in ambient (outside the device), n2 is the index of refraction in the device, lambda is the source wavelength, AR.INDEX is the user specified index of refraction of the AR coating, AR.THICK is the user defined thickness of the AR coating. In this case, the reflection coefficient of the coating is given by Equation 8-15, where theta is defined below.

$$R = \frac{\mathtt{AR.INDEX}^2(n_1 - n_2)\cos^2(\theta) + (n_1 n_2 - \mathtt{AR.INDEX}^2)\sin^2(\theta)}{\mathtt{AR.INDEX}^2(n_1 - n_2)\cos^2(\theta) + \left(n_1 n_2 - \mathtt{AR.INDEX}^2\right)\sin^2(\theta)} \qquad 8\text{-}9$$

$$\theta = \frac{2\pi \cdot \mathit{AR.INDEX} \cdot \mathit{AR.THICK} \cdot \cos\phi}{\lambda} \qquad 8\text{-}10$$

The parameters AR.INDEX and AR.THICK are defined in the INTERFACE statement and ϕ is the angle of incidence. The location of the interface must also be specified by the P1.X, P2.X, and P2.Y parameters of the INTERFACE statement.The syntax:

```
INTERFACE AR.INDEX=2.05 AR.THICK=0.07 P1.X=0.0 P1.Y=0.0 P2.X=10.0 P2.Y=0.0
```

defines a 70nm layer of real refractive index 2.05 at Y=0.0μm in a structure. The AR layer is assumed to be non-absorbing, that is the imaginary refractive index is zero.

For the case of non-normal incidence, absorbing AR coatings, or multi-layer AR coatings, the user must use the C-interpreter function F.REFLECT specified on the BEAM statement. Using this function,

the user can specify the reflection coefficient, angle of transmission, and transmitted polarization as a function of position, wavelength, angle of incidence, and incident polarization.

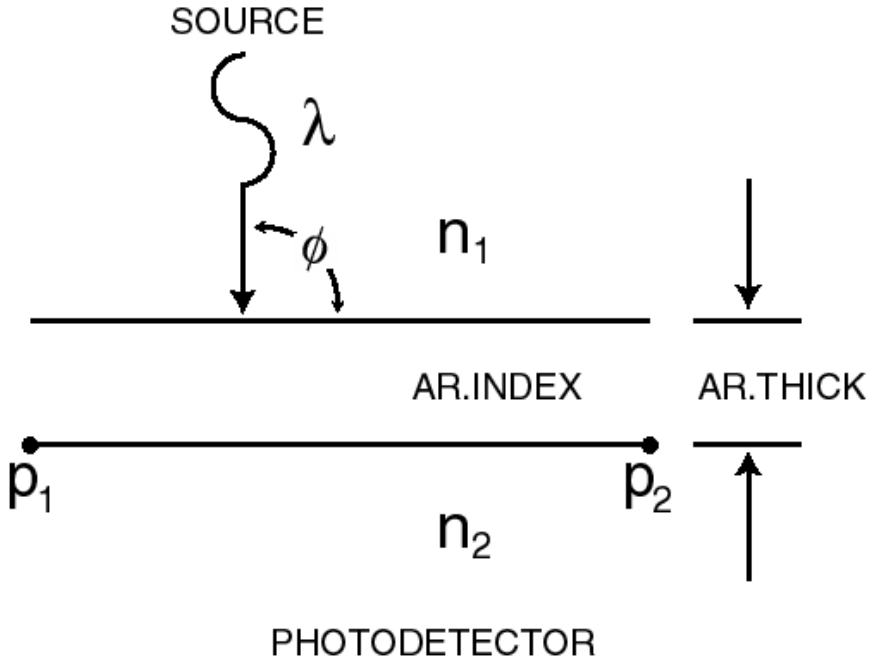

Figure 8-4: Single Layer AR Coating Under Normal Incidence

Light Absorption and Photogeneration

The cumulative effects of the reflection coefficients, transmission coefficients, and the integrated loss due to absorption over the ray path are saved for each ray. The generation associated with each grid point can be calculated by integration of the generation rate formula (Equation 8-11) over the area of intersection between the ray and the polygon associated with the grid point.

$$G = \eta_0 \frac{P^* \lambda}{hc} \alpha e^{-\alpha y} \qquad 8\text{-}11$$

where:

P* contains the cumulative effects of reflections, transmissions, and loss due to absorption over the ray path.

η_o is the internal quantum efficiency which represents the number of carrier pairs generated per photon observed.

y is a relative distance for the ray in question.

h is Planck's constant

λ is the wavelength.

c is the speed of light.

α is given by Equation 8-12.

$$\alpha = \frac{4\pi}{\lambda}k \qquad 8\text{-}12$$

where:

α is the absorption coeficient.

λ is the wavelength.

k is the imaginary part of the optical index of refraction.

Photogeneration on a Non-uniform Mesh

The photogeneration algorithm used integrates the optical intensity around each node point. This is done to ensure that the total photogeneration rate is not grid sensitive. A uniform photogeneration rate is defined as a constant value of (photogeneration rate at any node)*(element area around the node). In TONYPLOT a uniform photogeneration rate may appear to vary across a non-uniform mesh density.

Photogeneration at Contacts

The photogeneration associated with nodes that are also defined as electrodes is a special case. The electrical boundary conditions require that the carrier concentration at electrode nodes equals the doping level. This means that photogeneration at nodes which are electrodes must be zero. However just seting these nodes to zero photogeneration will typically cause an apparent drop in quantum efficiency.

The photogeneration rate at the contact nodes is calculated as usual. However this photogeneration rate is applied to the neighboring node inside the semiconductor. This means for a uniform mesh and photogeneration rate, if the photogeneration rate is $1.0x10^{17}$ pairs/cm^{-3}s, then the nodes at the contacts will have zero photogeneration and the next node into the semiconductor will have $2.0x10^{17}$ pairs/cm^{-3}s.

User-Defined Arbitrary Photogeneration

An option exists for the user to define the photogeneration rate. A C-Interpreter function written into a text file can be supplied to the program using the F.RADIATE parameter of the BEAM statement. For example if a file myoptics.c was developed using the template C-interpreter functions supplied it can be referenced. using:

```
BEAM NUM=1 F.RADIATE=myoptics.c
.
.
SOLVE B1=1.0
SOLVE B1=2.0
```

The file myoptics.c returns a time and position dependent photogeneration rate to the program. This returned value is mutiplied at every node point by the value of B1.

With this option all other parameters of the BEAM statement and all the material refractive indices are overridden.

Photocurrent and Quantum Efficiency

One of the important figures of merit of a photodetector is quantum efficiency. Here quantum efficiency is defined as the ratio of the number of carriers detected at a given photodetector electrode

divided by the number of incident photons on the detector. Ideally, this ratio should 1.0 in detectors without any positive feedback mechanisms, such as avalanche gain. LUMINOUS does not directly calculate quantum efficiency, but does calculate two useful quantities printed to the run-time output and saved to the log file. These quantities are *source photo-current* and *available photo-current* and can be viewed in TONYPLOT from log files produced using LUMINOUS.

Definition of Source Photocurrent

The source photocurrent for a monochromatic source is given in the equation below. Here B_n is the intensity in beam number *n* set by the user on the SOLVE statement. λ is the source wavelength specified by the WAVELENGTH parameter of the BEAM statement, h is Planck's constant, c is the speed of light, and W_t is the width of the beam including the effects of clipping (see section on Ray Tracing). This can be considered as a measure of the rate of photons incident on the device expressed as a current density.

$$I_s = q\frac{\boldsymbol{B_n}\lambda}{hc}W_t \qquad \text{8-13}$$

Definition of Available Photocurrent

The available photo-current for a monochromatic source is given by the equation below. Here all the terms in front of the sumation have the same definitions as for the source photo-current. The sum is taken over the number of rays traced, NR. W_R is the width associated with the ray. The integral is taken over the length, Yi, associated with the ray. P_i accounts for the attenuation before the start of the ray due to non-unity transmission coefficients and absorption prior to the ray start. And α_i is the absorption coefficient in the material that the ray is traversing. The available photo-current can be thought of as a measure of the rate of photo absorption in the device expressed as a current density. This should be similar but somewhat less than the source photo-current. The losses are due to reflection and transmission of light out of the device structure.

$$I_A = q\frac{\boldsymbol{B_n}\lambda}{hc}\sum_{i=1}^{N_R} W_R\int_o^{x_i} P_i\alpha_i e^{-\alpha_i y}dy \qquad \text{8-14}$$

Depending how the user wants to define it, quantum efficiency can be readily calculated by dividing the current from one of the device electrodes by either the "source photo-current" or the "available photo-current".

The definitions for "source photo-current" and "available photo-current" for multi-spectral sources are given in Equations 8-15 and 8-16. Here the symbols have the same definitions as in Equations 8-13 and 8-14, but the first summation is taken over the number of descrete wavelengths, N. P_j is the relative intensity at the wavelength, $lambda_j$.

$$I_s = q\frac{B_n}{hc}W_t\sum_{i=1}^{N_\lambda} P_i\lambda_i \qquad \text{8-15}$$

$$I_A = q\frac{B_n}{hc}\sum_{i=1}^{N_\lambda} P_i\lambda_i \sum_{i=1}^{N_R} W_R\int_o^{x_i} P_i\alpha_i e^{-\alpha_i y}dy \qquad \text{8-16}$$

Simulating Photodetectors

Overview

This section describes techniques to simulate photodetectors. This section applies to the simulation of any of the following devices: p-n and p-i-n photodiodes, avalanche photodiodes, Schottky photodetectors, CCDs, MSMs, photoconductors, optical FETs and optically triggered power devices.

Defining Optical Sources

Identifying an Optical Beam

Up to ten optical sources are allowed. Optical sources are described using the BEAM statement. All BEAM statements must appear somewhere after the MESH, REGION, DOPING, and ELECTRODE statements and before any SOLVE statement. The parameters of the BEAM statement describe a single optical source.

The NUM parameter is used to uniquely identify one optical source. Values between 1 and 10 are valid for the NUM parameter. Optical sources are subsequently referred to by the value of their NUM parameter. The power of the optical beam is set using the B<**n**> parameter of the SOLVE statement where **n** is the beam number defined by NUM.

Origin Plane of the Beam

The origin of the optical source is specified using the X.ORIGIN and Y.ORIGIN parameters. These parameters describe the origin of the optical beam relative to the device coordinate system. Currently, it is required that the origin lie outside any device region. The ANGLE parameter specifies the angle of the direction of propogation of the beam with respect to the device coordinate system. ANGLE=90 specifies vertical (normal) illumination from above.

The width of the optical beam is specified using the MIN.WINDOW and MAX.WINDOW parameters. These parameters specify the limits of the source beam relative to the beam origin. If either of the limits are omitted, that limit will be clipped to the edge of the device domain.

Note: It is extremely important that no section of the origin plane of the beam intersects or is inside the simulation grid otherwise incorrect results will be obtained. This is important to check in cases when ANGLE is not 90^{o} or 270^{o}.

Reflections

The user can also specify whether to ignore the first reflection using the FRONT.REFL and the backside and sidewall reflection using the BACK.REFL. It is useful to turn on the backside reflections for devices which use a back side reflector to improve collection efficiency. The number of reflections solved is set by the REFLECTS parameter.

Typically, BACK.REFL should be used if the structure simulated is equivalent to the complete photodetector geometry as in a descrete device. If the simulation structure is a section of a larger substrate as in CCD simulation then BACK.REFL should not be used.

Since reflection and transmission coefficients are used in the ray trace, for arbitrary angles of incidence, the user should also specify the polarization using the POLARIZATION parameter.

In complex structures it is useful to limit the ray tracing to trace only those rays with significant optical power. The parameter MIN.POWER is used to terminate ray traces that drop below MIN.POWER* (optical source power).

Monochromatic or Multispectral Sources

The optical source can be either monochromatic or multispectral. For monochromatic sources, the WAVELENGTH parameter can be used to assign the optical wavelength. WAVELENGTH uses the units microns to be more consistent with the rest of ATLAS. Users used to the optoelectronic engineering preference for nanometers should note this.

For multispectral sources spectral intensity is described in am external ASCII file, specified by the POWER.FILE parameter. This is a text file that contains a list of pairs defining wavelength and spectral intensity. The first line of the file gives the integer number of wavelength-intensity pairs in the file. An example of the contents of such a file is shown below.

```
4
0.4 0.5
0.5 1.0
0.6 1.2
0.7 1.1
```

This example specifies that there are four samples, and that at a wavelength of 0.4 μm the intensity is 0.5 Watts per square cm per μm of wavelength, and so on. With multispectral sources, the user must specify a discretization of the interpolated information. Values must be specified for the WAVEL.START, WAVEL.END and WAVEL.NUM parameters. These specify the starting and ending wavelengths and the number of wavelengths to sample. LUMINOUS will use wavelengths at equal intervals over the specified range of wavelengths.

The program performs an independent ray trace at each of the sample wavelengths. For example:

```
WAVEL.START=0.4 WAVEL.END=0.6 WAVEL.NUM=2
```

will cause ray traces at wavelengths of 0.45 and 0.55. LUMINOUS obtains the intensity associated with each sample by integrating the values of the spectral intensity file using a piece wise linear approximation. Each integral is performed over the range between succesive midpoints. In the preceeding example, the integration for the sample at 0.45 would be performed over the range of 0.4 to 0.5.

For either the monochromatic or multispectral sources the wavelength(s) and intensities can be uniformly scaled using the WAVEL.SCALE parameter and the POWER.SCALE parameter respectively. These parameters are useful if the intensities or wavelengths are specified in units other than the default units.

Note: The units of spectral intensity in the POWER.FILE are W/cm^2 per μm of wavelength. The per micron of wavelength is important to remember when integrating the total power across the spectrum.

Defining Optical Properties of Materials

For ray tracing the complex index of refraction of the various material regions in the structure must be specified. For many of the more common semiconductors and insulators, there are built in tables of index versus wavelength. For those materials lacking reasonable default complex index of refraction, the user can specify the index.

Note: The parameter INDEX.CHECK can be added to any SOLVE statement to print out the refractive indices being used for that bias step.

Setting Single Values For The Refractive Index

The parameters REAL.INDEX and IMAG.INDEX of the MATERIAL statement can be used to set the real and imaginary indices respectively of a specified material, region or regions.

For example, the statement:

```
MATERIAL MATERIAL=Air REAL.INDEX=1.0 IMAG.INDEX=0.0
```

would set the index for all material regions composed of "Air".

The statement:

```
MATERIAL REGION=1 REAL.INDEX=1.0 IMAG.INDEX=0.0
```

would set the index for region number 1.

The statement:

```
MATERIAL REAL.INDEX=1.0 IMAG.INDEX=0.0
```

would set the index for all regions.

Setting A Wavelength Dependent Refractive Index

The preceeding examples set the complex index of refraction for a material regardless of wavelength. This is probably adequate for monochromatic sources. For multispectral simulations, the index of refraction should be modeled as having a dependence on wavelength. There are two ways to do this.

ASCII File Input

The first way is to specify index versus wavelength in a file. This is a text file that contains ordered triplets of wavelength, real index, and imaginary index. The first entry in the table is the number of samples. If the INDEX.FILE parameter of the MATERIAL statement is set to the name of the index file, Linear interpolation from this table will be done to obtain the index of refraction as a function of wavelength.

A valid index file is shown below.

```
2
0.5 2.0 0.0
0.6 3.0 0.02
```

In this example, a real index of 2.5, and an imaginary index of 0.015 would be used for a wavelength of 0.55 microns.

C-Interpreter Function

The dependence of complex index on wavelength can also be specified using the C-Interpreter function. The syntax is:

```
MATERIAL NAME=Silicon F.INDEX=myindex.c
```

The file myindex.c is an external file conforming to the template supplied with the program. It returns wavelength dependent real and imaginary indices. Instructions for using the C-Interpreter and finding the template functions are described in Appendix A.

Extracting Dark Characteristics

One of the first tasks in analyzing a new detector design is to examine dark current, device capacitance, and possibly other un-illuminated characteristics. This can normally be done without the use of LUMINOUS. Extraction of the characteristics is adequately covered in the chapters on S-PISCES or BLAZE.

The extraction of reverse bias leakage currents for diodes presents some difficult numerical problems for device simulators. These problems save associated with limitations on numerical precision. ATLAS, as well as most other available device simulators, uses double precision arithmetic to evaluate terminal currents. Double precision arithmetic provides roughtly 16 decimal digits of precision. With the scaling performed internally, this allows the measurement of currents down to a level of between about 10^{-12} A/micron to 10^{-16} A/micron. Unfortunately, photodiode leakage currents are often around or below this level. This means that the currents printed are subject to significant numerical noise, and do not provide an accurate estimate of the device leakage currents. Two ways of estimating reverse leakage current are available to the user.

Integrated Recombination

From a theoretical standpoint, the reverse behavior of diodes can be dominated by one of two effects: diffusion currents in the neutral regions or recombination currents inside the depletion region[13]. ATLAS can provide insight into both of these contributing mechanisms. To estimate recombination current, use the MEASURE statement to calculate the integrated recombination rate. The following statement can be used:

```
MEASURE U.TOTAL
```

When this statement is executed, it prints out the total integrated recombination rate. The user needs to multiply this value by the electron charge ($1.0623*10^{-19}$ coulombs) to obtain an estimate of the recombination current contribution to the reverse diode leakage current.

Extrapolation from High Temperatures

The diffusion current contribution can be estimated by taking advantage of the non-linear relationship between the diffusion current and temperature. Referring to the expression for the "Ideal Diode" current given by Equation 8-17, the dominant temperature dependency arises from the variation of the intrinsic concentration.

$$J = \left(\frac{qD_p p_{n0}}{L_p} + \frac{qD_n n_{p0}}{L_n}\right)\left(\exp\left(\left(\frac{qV}{kT_L}\right) - 1\right)\right) \qquad 8\text{-}17$$

where n_{p0} and p_{n0} are thermal-equilibrium minority carrier densities on either side of the junction. This gives an exponential variation of the diffusion current with temperature as given in Equation 8-18:

$$J \approx \exp\left(\frac{-E_g}{kT_L}\right) \cdot \left(\exp\left(\frac{qV}{kT_L}\right) - 1\right) \qquad 8\text{-}18$$

This relation can be used to estimate the diffusion current contribution at the operating temperature. The basic idea is to calculate the current at a high temperature where the problem of numerical precision does not arise, and then scale the current to the operating temperature using Equation 8-12. For example, if the device is to operate at 300K, the temperature may be set to 450K using the TEMPERATURE parameter of the MODEL statement. Any temperature dependence of the energy gap should be disabled by explicitly specifying the band gap using the EG300 parameter and setting

EGALPHA and EGBETA parameters to zero, all on the MATERIAL statement. The following statement illustrates this approach as it might apply to a silicon diode:

```
MODEL TEMPERATURE=450
MATERIAL EG300=1.12 EGALPHA=0.0 EGBETA=0.0
```

ATLAS can then be used to obtain the reverse bias current at the elevated temperature. The following equation can be applied to obtain the depletion current contribution at the operating temperature:

$$J = J_e \cdot \exp\left(\frac{E_g}{kT_e} - \frac{E_g}{kT_L}\right) \cdot \left(\frac{\exp\left(\frac{qV}{kT_L}\right) - 1}{\exp\left(\frac{qV}{kT_e}\right) - 1}\right) \quad \text{8-19}$$

where J_e is the current measured at the elevated temperature, E_g is the bandgap, T_e is the elevated temperature, T_L is the operating temperature, V is the operating bias voltage, and J is the current estimate at the operating temperature. Once the user has obtained estimates of the recombination and diffusion contributions, the total leakage current can be obtained by summing the two contributions

Numerical Solution Parameters

ATLAS uses a cut-off value of carrier concentration below which solutions are not required to converge. This limit is set by the parameter CLIM.DD. See the 'Numerical Methods' chapter for more detail on CLIM.DD. For photodetectors it is often necessary to reduce CLIM.DD to 10^5 in order to resolve carrier concentrations in depleted regions before illumination.

Extracting Detection Efficiency

One of the simpler tasks in characterizing a photodetector design is to measure DC detection efficiency. This will be typically done as a function of bias voltage, optical intensity, or wavelength. Each of these analyses can be performed using the SOLVE statement. The Bn parameter can be used to set the optical intensity of the optical sources described in the previous section. The following example illustrates obtaining a solution with a specified optical intensity:

```
SOLVE B1=1.0
```

This specifies that a solution is to be obtained for an optical intensity in the beam numbered "1" of 1.0 Watt/cm^2. If this were the first SOLVE statement specified, the ray trace in LUMINOUS would be initiated. At the start of the solution the optical intensities of each optical source with a positive intensity is printed. In addition the available photo-current and source photo-current are printed. See the prior section on Photocurrent for a definition of these two quantities.

Internal and External Quantum Efficiency

The available photocurrent divided by the source photocurrent is a measure of the external quantum efficiency of the detector.

The calculated terminal current can be divided by the source or available photocurrents is used to evaluate the internal quantum efficiency of the device.

Obtaining Quantum Efficiency versus Bias

The intensities specified in the SOLVE statement apply until another SOLVE statement changes the intensity of the beam. Sequences of SOLVE statements can be used to vary the the optical intensity at arbitrary intervals. The simple linear ramps of optical intensity can be abbreviated using the LIT.STEP and NSTEP parameters of the SOLVE statement. The LIT.STEP parameter specifies the size of the DC step and NSTEP specifies how many steps are desired.

Another option for analyzing DC quantum efficiency is to fix the optical intensity and vary bias voltages. The bias voltages can be varied in arbitrary discrete steps using several SOLVE statements, or in a linear ramp using individual SOLVE statements. This is useful for determining the optimum operating bias of devices such as avalanche detectors and photo transistors.

Obtaining Transient Response to Optical Sources

It is sometimes desirable to examine the time domain response of a detector to time-dependent (e.g. ramped or pulsed) optical sources. LUMINOUS provides this capability with the RAMP.LIT parameter. When the RAMP.LIT parameter is specified in a SOLVE statement, the optical intensity is changed linearly from the most recently set intensity to the intensity set in the B parameter. If a particular source intensity is not set using the corresponding B parameter its intensity is not varied during the transient.

The period of the linear ramp is specified by the RAMPTIME parameter. The transient simulation stops after the time specified by the TSTOP parameter. If the time given by TSTOP is greater than that given by RAMPTIME, the source intensities remain constant until the time given by TSTOP. For transient ramps the TSTEP parameter should also be set. TSTEP is typically set to allow several samples within the RAMPTIME. After the first time step, subsequent time step sizes will be chosen automatically based on estimates of truncation error. The following is an example of the specification of an optical impulse transient:

```
SOLVE B1=1.0 RAMPTIME=1E-9 TSTOP=1E-9 TSTEP=1E-11
SOLVE B1=0.0 RAMPTIME=1E-9 TSTOP=20E-9 TSTEP=1E-11
```

In this example a triangular impulse in the intensity of the optical source is simulated. The peak intensity is 1.0 and the impulse width is 2 ns. The response of the device is simulated for an additional 18 ns. An initial time step of 10 ps is chosen for both parts of the impulse.

Obtaining Frequency Response to Optical Sources

Small signal response to optical sources can also be simulated . To obtain a solution for small signal response the SS.PHOT parameter should be set in the SOLVE statement. The BEAM parameter must also be asigned to the specific optical source index for small signal response of that source. A single source small signal frequency can be specified using the FREQUENCY parameter. The frequency can be varied within a single SOLVE statement using the NFSTEP and FSTEP parameters. The NFSTEP indicates how many frequency steps are to be simulated; while, the FSTEP indicates the step size. If the MULT.F parameter is specified the start frequency is multiplied by the step size for the specified number of steps. Otherwize, the step size is added to the start frequency for the specified number of steps.

For example:

```
SOLVE SS.PHOT BEAM=1 FREQUENCY=1e6 NFSTEP=5 FSTEP=10 MULT.F
```

will invoke solutions as optical frequencies at every decade from 1MHz to 100 GHz.

If the small signal parameters are specified in the same SOLVE statement as a DC bias ramp the small signal response is extracted for each bias voltage for each small signal frequency. This is a useful strategy for analyzing the frequency response of the device as a function of bias voltage.

Obtaining Spatial Response

To obtain spatial response, an optical spot is moved along a line segment perpendicular to the direction of propagation of the source. Each incremental step is equal to the width of the spot. The total distance over which the source is scanned is defined by the MIN.WINDOW and MAX.WINDOW parameters of the BEAM statement. The number of steps is defined by the RAYS parameter of the BEAM statement. The spot width is defined by the ratio:

```
(MAX.WINDOW - MIN.WINDOW)/RAYS
```

The spot scan is started by the SCAN.SPOT parameter of the SOLVE statement. This parameter is set to the beam index of the optical source to be scanned (i.e. the beam defined by the BEAM statement whose NUMBER parameter is set to the beam index). During the spot scan ATLAS obtains solutions and outputs terminal currents, etc. as well as the relative beam location at each incremental spot location. This information can be used by **TonyPlot** to produce plots of photresponse as a function of position.

Obtaining Spectral Response

The spectral response, defined as device current as a function of the wavelength of the optical source wavelength, can be obtained. The LAMBDA parameter of the SOLVE statement sets the source wavelength of the beam in microns. Since the wavelength can be set in each SOLVE statement, successive solutions can be obtained as a function of wavelength. Each time the LAMBDA parameter is specified, a new ray trace is run for that new wavelength and the electrical solution recalculated.

The following statements could be used to extract terminal currents at a series of discrete wavelengths:

```
SOLVE B1=1 LAMBDA=0.2
SOLVE B1=1 LAMBDA=0.3
SOLVE B1=1 LAMBDA=0.4
SOLVE B1=1 LAMBDA=0.5
SOLVE B1=1 LAMBDA=0.6
SOLVE B1=1 LAMBDA=0.7
SOLVE B1=1 LAMBDA=0.8
```

In this example, the spectral response is obtained for wavelengths from 0.2 microns to 0.8 microns.

When using LAMBDA the WAVELENGTH parameter of the BEAM statement is overridden. However users should be sure to use a monochromatic beam and not a multi-spectral beam for this simulation..

Note: The units of LAMBDA and WAVELENGTH in the BEAM statement are in μm.

Simulating Solar Cells

Obtaining Open Circuit Voltage and Short Circuit Current

To obtain V_{OC} and I_{SS} for a solar cell, first the illumination conditions should be defined. This should be done as discussed above in describing multi-spectral sources. The short circuit current is obtained by defining the contacts as voltage dependent contacts (default) and obtaining a solution with the device under zero bias with illumination. This can typically be done as a first step by the following input statement, for example:

```
SOLVE B1=1
```

In this statement it is assumed the source B1 has been already defined and the intensity is that which is expected in the actual device. When a solution is obtained the terminal currents represent the short circuit current.

The open circuit voltage is obtained by defining one or more of the contacts as current controlled. This is done using the CONTACT statement. For example:

```
CONTACT NAME=anode CURRENT
```

defines that electrode number 1 is a current controlled electrode.The open circuit voltage is then obtained by setting the current at this contact to zero, and obtaining a solution. The following statement illustrates this:

```
SOLVE I1=0.0 B1=1
```

Once the solution is obtained, the bias associated with I1 is the open circuit voltage.

Simulating LEDs

LUMINOUS used with BLAZE can be used to extract certain parameters associated with light emitting devices. Extraction of the integrated radiative recombination in the device provides an estimate of luminous intensity, and extraction of integrated total recombination rate allows an estimate of luminous efficiency.

To calculate luminous efficiency, all recombination models should be enabled using the appropriate MODEL and MATERIAL statements. The device should be biased to its operating point. The MEASURE statement should be used to obtain the integrated radiative recombination rate using the U.RADIATIVE parameter. Then the MEASURE statement should be used to obtain the integrated total recombination rate using the U.TOTAL parameter. A ratio of radiative recombination rate with total rate is an estimate of luminous efficiency. The following illusates this procedure.

```
MEASURE U.RADIATIVE
MEASURE U.TOTAL
```

In order to extract luminous intensity a luminous wavelength must be specified by the user. This is done by setting the L.WAVE parameter of the SOLVE statement. When the luminous wavelength is set, the program automatically calculates the luminous intensity of the device and stores this data in the log file. The following SOLVE statement illustrates this process.

```
SOLVE L.WAVE=0.8
```

The equation for luminous intensity (P) is:

$$P = \frac{h_C}{\mathbf{L.WAVE}} \int R_{RAD}\, dA \qquad \text{8-20}$$

Chapter 9: Laser

Introduction

LASER is an ATLAS product that performs coupled electrical and optical simulation of semiconductor lasers. LASER works in conjunction with BLAZE and allows you to:

Solve the Helmholtz equation, in order to calculate the optical field and photon densities.

- Calculate the carrier recombination due to light emission (i.e. stimulated emission).
- Calculate optical gain, which may depend on the photon energy and the quasi-Fermi level.
- Calculate laser light output power.
- Calculate the light intensity profile corresponding to the fundamental transverse mode.
- Calculate light output and modal gain spectra for several longitudinal modes.

A basic familiarity with ATLAS and BLAZE is assumed. You should read the Getting Started and BLAZE chapters before reading this chapter.

Physical Models

In order to simulate semiconductor lasers, the basic semiconductor equations (Equations 3-1 to 3-4 in Chapter 3 of this manual) are solved self-consistently with an optical equation that determines the optical field intensity distribution. LASER uses the following coordinate system:

- the x axis is perpendicular to the laser cavity and goes along the surface (from left to right).
- the y axis is perpendicular to the laser cavity and goes down from the surface.
- the z axis goes along the laser cavity.

The x and y axes are the same as in other ATLAS products. The electrical and optical equations are solved in the x, y plane (i.e. perpendicular to the laser cavity).

LASER solves a two-dimensional Helmholtz equation to determine the transverse optical field profile E(x,y):

$$\nabla_{xy}^2 E(x, y) + \left(\frac{\omega_m^2}{c^2} \varepsilon(x, y) - \beta^2 \right) E(x, y) = 0 \qquad 9\text{-}1$$

where:

$\nabla_{xy}^2 = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right)$ is the two-dimensional Laplace operator,

ω_m is the frequency corresponding to longitudinal mode m (ω_m corresponds to LAS.OMEGA on the MODELS statement for single frequency simulations),

c is the velocity of light in vacuum,

$\varepsilon(x,y)$ is the high frequency dielectric permittivity.

Equation 9-1 is a complex eigenvalue problem. LASER solves this equation to determine the set of complex eigenvalues β_k and corresponding eigenfunctions $E_k(x,y)$. LASER takes into account only the fundamental transverse mode solution, so the index k will be dropped from subsequent equations.

In principle, equation 9-1 should be solved for each longitudinal mode that is taken into account. Since very few longitudinal modes are actually lasing, LASER equation 9-1 only once, for the longitudinal

mode with the greatest power, and subsequently assumes $E_m(x,y)=E_o(x,y)$, where $E_o(x,y)$ is the optical field corresponding to the most powerful longitudinal mode. This assumption is reasonable, since the shape of the solution is almost independent of frequency within the range of interest.

For dielectric permittivity, LASER uses the following model [135]:

$$\varepsilon(x, y) = n_0^2 + (-\mathrm{ALPHAR} + j)\frac{n_0 g(x, y)}{k_\omega} - j\frac{n_0(\mathrm{ALPHAA} + \mathrm{FCN} \cdot n + \mathrm{FCP} \cdot p)}{k_\omega} \qquad 9\text{-}2$$

where:

- n_0 is the bulk refractive index
- ALPHAR is a line width broadening factor,
- $j = \sqrt{-1}$,
- $k\omega = \omega/c$, and:
- g (x,y) is the local optical gain
- ALPHAA is the bulk absorption loss and is specified on the MATERIAL statement.
- LAS.ABSORPTION must be specified on the MODELS statement to include absorption loss.
- FCN and FCP are the coefficients of the free-carrier loss and are set via the MATERIAL statement. LAS.FCARRIER must be specified on the MODELS to include this loss mechanism.

Table 9-1: User Specifiable Parameters for Equation 9-2

Statement	Parameter	Default	Units
MATERIAL	ALPHAR	4.0	
MATERIAL	ALPHAA	0.0	
MATERIAL	EPSINF		
MATERIAL	FCN	3.0e-18	cm
MATERIAL	FCP	7.0e-18	cm
MODELS	LAS.ABSORPTION	FALSE	
MODELS	LAS.FCARRIER	FALSE	

Local Optical Gain

The central model in semiconductor laser simulation is the optical gain model. The optical gain is the ability of the semiconductor media to amplify light, i.e. the number of photons generated per unit length. LASER provides two models for local gain. The first model is physically based. It takes into account frequency dependence, and can be used for spectral analysis [97, 98]. The optical gain using this model is given by:

$$g(x, y) = GAIN0 \sqrt{\frac{h\omega - E_g}{kT}} \left[f\left(\frac{E_c - E_{fn} + GAMMA(h\omega - E_g)}{kT} \right) - f\left(\frac{E_v - E_{fp} - (1 - GAMMA)(h\omega - E_g)}{kT} \right) \right] \quad 9\text{-}3$$

Table 9-2: User Specifiable Parameters for Equation 9-3

Statement	Parameter	Default	Units
MATERIAL	GAIN0	2000.0	cm^{-1}
MATERIAL	GAMMA		

where:

E_{fn} and E_{fp} are the quasi-Fermi levels for electrons and holes respectively,

E_g is the band gap, E_c is the conduction band edge energy, E_v is the valence band edge energy, and:

$$f(x) = \frac{1}{\exp(x) - 1} \quad 9\text{-}4$$

The user may specify the value of the parameter GAMMA as shown in Table 9-2. If the parameter GAMMA is not specified, its value will be calculated as in Equation 9-5.

$$GAMMA = \frac{1}{\left(\frac{N_c}{N_v} \right)^{\frac{2}{3}} + 1} \quad 9\text{-}5$$

where N_c and N_v are the material specific densities of states in the conduction band and the valence band respectively.

The second model is a simple empirical model that is defined by the following equation:

$$g(x, y) = GAIN00 + GAIN1N \cdot n + GAIN1P \cdot p + GAIN2NP \cdot np + GAIN1MIN \cdot \min(n, p) \quad 9\text{-}6$$

Table 9-3: User Specifiable Parameters for Equation 9-6

Statement	Parameter	Default	Units
MATERIAL	GAIN00	-200.0	cm^{-1}
MATERIAL	GAIN1N	0	cm^2

Table 9-3: User Specifiable Parameters for Equation 9-6

Statement	Parameter	Default	Units
MATERIAL	GAIN2N	0	cm^2
MATERIAL	GAIN1NP	0	cm^5
MATERIAL	GAIN1MIN	3.0e-16	cm^2

This model does not take into account frequency dependence and is valid only for the lasing frequency. It can not be used for calculations that involve multiple longitudinal modes.

Stimulated Emission

Carrier recombination due to stimulated light emission is modeled as follows:

$$R_{st}(x, y) = \sum_{m} \frac{c}{LAS.NEFF} g(x, y)|E(x, y)|^2 \cdot S_m \quad 9\text{-}7$$

where R_{st} is the recombination rate due to stimulated light emission, LAS.NEFF is the group effective refractive index and S is the photon density. This subscript m in this and all subsequent equations refers to a modal quantity, for example S_m in Equation 9-7 is the photon density for mode m. The parameter LAS.NEFF is user specifiable as shown in Table 9-4.

Table 9-4: User Specifiable Parameters for Equation 9-7

Statement	Parameter	Default	Units
MODELS	LAS.NEFF	3.57	

Photon Rate Equations

Linkage between optical and electrical models is provided by the optical gain. The optical gain depends on the quasi-Fermi levels and in turn impacts dielectric permittivity (see Equation 9-2); and by the coupling between the stimulated carrier recombination rate (R_{st}) and the density of photons S, as described by Equation 9-7. To determine S_m, LASER solves the system of photon rate equations:

$$\frac{dS_m}{dt} = \left(\frac{c}{LAS.NEFF} G_m - \frac{1}{\tau_{ph_m}} - \frac{c \ LAS.LOSSES}{LAS.NEFF}\right) S_m + R_{sp_m} \quad 9\text{-}8$$

where the modal gain G_m is given by:

$$G_m = \iint g_m(x, y) \cdot |E(x, y)|^2 \cdot dx \ \ dy \quad 9\text{-}9$$

and the modal spontaneous emission rate R_{sp_m} is given by:

$$R_{sp_m} = \iint (r_{sp}(x, y))_m \cdot |E(x, y)|^2 \cdot dx \ \ dy \quad 9\text{-}10$$

LAS.LOSSES is the internal losses and can be specified on the MODEL statement as shown in Table 9-5. E(x,y) is the normalized optical field.

Table 9-5: User Specifiable Parameters for Equation 9-8

Statement	Parameter	Default	Units
MODELS	LAS.LOSSES	0	cm^{-1}

The modal photon lifetime in Equation 9-8, τ_{ph_m}, represents the losses in the laser. The losses per mode are given by[96]:

$$\frac{1}{\tau_{ph}} = \frac{c}{LAS.NEFF}(\alpha_a + \alpha_{fc} + \alpha_{mir}) \quad 9\text{-}11$$

α_a is the bulk absorption loss, α_{fc} is the free-carrier loss and α_{mir} is the mirror loss. These are defined as:

$$\alpha_a = \iint ALPHAA \cdot |E(x, y)|^2 dx\, dy \quad 9\text{-}12$$

$$\alpha_{fc} = \iint (FCN\ n + FCP\ p) \cdot |E(x, y)|^2 \cdot dx\, dy \quad 9\text{-}13$$

$$\alpha_{mir} = \frac{1}{2\ CAVITY.LENGTH} \ln\left(\frac{1}{MIRROR.LOSS^2}\right) \quad 9\text{-}14$$

Table 9-6: User Specifiable Parameters for LASER Loss Models

Statement	Parameter	Default	Units
MODELS	MIRROR.LOSS	90.0	%
MODELS	CAVITY.LENGTH	100.0	μm

The user specifiable parameters for the loss models are given in Tables 9-6 and 9-7. MIRROR.LOSS is the percentage reflectivity for the facet cavity mirrors, CAVITY.LENGTH is the length of the laser cavity.

$r_{sp_m}(x, y)$ is calculated as follows when the physically-based model is used for local optical gain:

$$r_{sp_m}(x, y) = g_o \frac{\sqrt{h\omega - E_g}}{kT} \cdot f\left(E_c - E_{fn} + GAMMA\frac{(h\omega - E_g)}{kT}\right) \cdot \left[1 - f\left(E_v - E_{fp} - \frac{(1 - GAMMA)(h\omega - E_g)}{kT}\right)\right] \quad 9\text{-}15$$

For the steady-state case, the time dependent terms in the photon rate equation (Equation 9-8) are set to zero.

Solution Techniques

LASER solves the electrical and optical equations self consistently. Two models are available. The first is a single-frequency model that does not take into account the existence of multiple longitudinal modes. It is in this sense similar to the work described in. [94 ,95, 96]. The second model accounts for multiple longitudinal modes, and is in this sense similar to the work described in [96].

LASER solves the Helmholtz equation in a rectangular region that is a subdomain of the general ATLAS simulation region. An eigenvalue solver provides a set of eigenfunctions and corresponding eigenvectors, and LASER selects the fundamental transverse mode solution.

LASER uses boundary conditions of the form E(x,y)=0 on the boundaries of the solution region. This region should therefore be large enough to cover the entire active region of the laser diode, with some inclusion of the passive regions. When the single frequency model is used for simulation, the lasing frequency used in Equation 9-1 is an external parameter that is fixed during the calculation, and either model for optical gain can be used.

The multiple longitudinal mode model requires the use of the physically-based optical gain model. The user specifies an initial estimate of the lasing frequency, and this is adjusted during calculations. The user also specifies a frequency range or photon energy range within which LASER will calculate multiple longitudinal modes.

Specifying Laser Simulation Problems

The structure of the laser diode and the mesh used to simulate it are specified in the normal way using the capabilities provided by ATLAS and BLAZE. To enable LASER simulation, the user must do the following (using the parameters as shown in Table 9-7):

- **Activate** LASER **–** Specifies the LASER parameter of the MODELS statement.
- **Define a mesh for solution of the Helmholtz equation** – The Helmholtz equation is solved on a uniform rectangular mesh that is independent of the triangular mesh used for device simulation. This rectangular mesh and the region mesh in which the Helmholtz equation is solved is specified using the LX.MESH and LY.MESH statements. The specified region must lie completely inside the ATLAS simulation domain, and should completely cover the actively lasing region.
- **Specify a gain model** – The optical gain model is specified using the GAINMOD parameter in the MODELS statement. GAINMOD=1 specifies the use of the physically based model, and GAINMOD=2 specifies the use of the simple empirical model. For single-frequency calculations you can select different gain models for different regions/materials using the REGION or MATERIAL parameters of the MODELS statement.

Note: If multiple longitudinal modes are to be accounted for, then GAINMOD=1 must be specified for the active lasing region.

- Specify laser physical parameters and models.
  - Specify CAVITY.LENGTH (MODELS statement) - the length of the laser cavity in the z direction.
  - Specify PHOTON.ENERGY or LAS.OMEGA to specify (initial) photon energy or laser frequency (MODELS statement).
  - Specify laser loss mechanisms (LAS.MIRROR, LAS.FCARRIER, LAS.ABSORPTION on the MODELS statement) and any associated constants (FCN, FCP, ALPHAA on the MATERIAL statement

– Specify any additional laser losses LAS.LOSSES, (MODELS statement)

- If calculation of the laser spectrum is desired, **specify the multiple longitudinal modes model and additional parameters**:

  – Specify LMODES in the statement (enables the multiple mode model),

  – Specify LAS.EINIT, LAS.EFINAL in the MODELS statement. These parameters set the photon energy range within which LASER will take into account multiple longitudinal modes. (Make sure that initial photon energy is within this range, and that is specified for the active lasing region).

  – Specify the photon energy separation (LAS.ESEP). If this is not specified, LASER will automatically calculate the number of longitudinal modes based on the cavity length and the energy range. It is recommended that LASER be allowed to choose the photon energy separation.

  – Specify spectrum file name (SPEC.SAVE on the MODELS statement). LASER will produce a structure file containing spectrum data after calculation of each bias point. LASER will automatically append _dcN.log to the specified file name (where N is the number of the bias point) for steady-state solutions or _trN.log for a transient simulation. The first bias point where LASER is active will have N=1. (This is often not the first bias point during simulation). These files can be examined using TONYPLOT. If the parameter LAS.SPECSAVE (MODELS statement) is specified, the spectrum files will only be saved on every las.specsave solution.

Note: The index in the spectrum file name will still increase by one each time. The spectrum data can be stored in a single file for the transient simulation (only if LAS.MULTISAVE - MODELS statement) is set to FALSE.

- **Saving near and far field patterns** – The user may optionally save near and far field patterns by specifying a value for the PATTERNS parameter on the SAVE statement. The value of the PATTERNS parameter is a character string representing the root name of a file for saving the near and far field patterns. The near field pattern is saved to a file with the string ".nfp" appended to the root and the far field pattern is saved to a file with the string ".ffp" appended to the root name. These files can be examined using TONYPLOT.
- **Optionally, specify numeric parameters** – The numeric parameters are LAS.TOLER, LAS.ITMAX, LAS.SIN, LAS.TAUSS, LAS.MAXCH. The default values of these parameters have been selected to give a good trade-off between accuracy, efficiency, and robustness for most applications. Users can "fine-tune" the calculation by specifying different values. These parameters are discussed below.

Numerical Parameters

The following numerical parameters control the way LASER simulation is performed. All of these parameters have reasonable default values, but the user can specify the values.

- LAS.TOLER - sets the desired relative tolerance of the photon density calculation. The default value is 0.01. Setting this parameter to a lower value may slow down the calculation significantly. Using a larger value will result in faster but less accurate calculations.
- LAS.ITMAX - set maximum number of external LASER iterations during photon density calculation. The default value is 30.
- LAS.SIN - This is the initial photon density used only with simple LASER models. LASER starts the iteration process for photon density calculation from this value. This parameter

influences only the calculation time for the first bias point after the laser threshold is reached.

- LAS.TAUSS - This is an iteration parameter used in the calculation of photon densities. Using a larger value of this parameter can speed up the calculation but may cause convergence problems.
- LAS.MAXCH - This is the maximum allowable relative change of the photon density between LASER iterations. Using a larger value of this parameter can speed up the calculation but may cause convergence problems.

Table 9-7: LASER Parameters

Statement	Parameter	Default	Units
MODELS	GAINMOD		
MODELS	CAVITY.LENGTH	100	cm
MODELS	PHOTON.ENERGY		
MODELS	LAS.OMEGA	2.16e15	Hz
MODELS	LAS.MIRROR	90	%
MODELS	LAS.FCARRIER	FALSE	
MODELS	LAS.ABSORPTION	FALSE	
MODELS	LAS.LOSSES	0	
MODELS	LAS.EINIT	0	
MODELS	LAS.EFINAL	0	
MODELS	SPEC.NAME	spectrum.log	
MODELS	LAS.SPECSAVE	1	
MODELS	LAS.MULTISAVE	TRUE	
MATERIAL	LAS.TOLER	0.01	
MATERIAL	LAS.ITMAX	30	
MATERIAL	LAS.SIN	100000	cm^2
MATERIAL	LAS.TAUSS	0.05	
MATERIAL	LAS.MAXCH	2.5	

Semiconductor Laser Simulation Techniques

The most common technique for simulating laser diodes is to simulate forward characteristics with gradually increasing bias. The forward voltage is usually specified, but current boundary conditions can be used, and external elements can be included.

To save computational time, we recommend that you do not enable LASER models at the start of the simulation. Ramp the device bias first, and then enable LASER simulation using the LASER parameter in an additional MODELS statement.

In general, the single frequency LASER model, which does not take into account the longitudinal mode spectrum, is faster. This model provides good results very reliably, and is recommended for use if the lasing spectrum is not the subject of interest.

If the multiple longitudinal mode model is used, computational time will be longer and strongly dependent on the number of longitudinal modes involved in the calculation. Although ATLAS is a two-dimensional simulation framework, the laser spectrum and all other laser results are strongly dependent on cavity length, which is effectively the device length in the z direction. This situation is different than for other ATLAS simulators.

For example, terminal currents can normally be scaled to apply to a different length in the third dimension by simple multiplication. This is not the case for LASER, since the cavity length will determine energy spacing between longitudinal modes and will influence all other laser characteristics. Be sure to specify the parameter CAVITY.LENGTH (in the MODELS statement) when the longitudinal mode spectrum is to be calculated.

[This page intentionally left blank]

Chapter 10: MIXEDMODE

Introduction

The MIXEDMODE Concept

MIXEDMODE is a circuit simulator that can include elements simulated using device simulation, as well as compact circuit models. It combines different levels of abstraction to simulate relatively small circuits where compact models for single devices are not available or sufficiently accurate. In addition MIXEDMODE allows the user to also do multi-device simulations. MIXEDMODE uses advanced numerical algorithms that are efficient and robust for DC, transient, small signal AC and small signal network analysis.

MIXEDMODE is typically used to simulate circuits that contain semiconductor devices for which accurate compact models do not exist or circuits in which devices that play a critical role must be modeled very accurately. Applications of MIXEDMODE include: power circuits that may include diodes, power transistors, IGBTs, and GTOs; optoelectronic circuits; circuits subject to single event upset, thin film transistor circuits, high-frequency circuits, precision analog circuits, and high performance digital circuits.

MIXEDMODE circuits can include up to 100 nodes, 300 elements, and up to ten numerical simulated ATLAS devices. These limits are reasonable for most applications however they can be increased in custom versions on request to Silvaco. The circuit elements that are supported include dependent and independent voltage and current sources, as well as resistors, capacitors, inductors, coupled inductors, MOSFETs, BJTs, diodes, and switches. Commonly used SPICE compact models are available; and the SPICE input language is used for circuit specification.

Organization of this Chapter

This chapter describes circuit simulation capabilities rather than device simulation capabilities. It is therefore organized differently than the chapters that describe other ATLAS products. The first part of the chapter contains introductory and background information. The middle section presents and explains MIXEDMODE syntax. This is followed by some sample input decks. The final sections contain a statement reference and a detailed description of the provided electrical compact models for diodes, BJTs, and MOSFETs.

Background

Circuit simulators such as SPICE solve systems of equations that describe the behavior of electrical circuits. The devices that are of interest to circuit designers are normally well characterized. 'Compact' or 'circuit' models are analytic formulae that approximate measured terminal characteristics. Advanced compact models provide high accuracy with minimum computational complexity. Device modeling, device characterization and parameter extraction are concerned with the development and use of accurate and efficient compact models.

Physically based device simulation solves systems of equations that describe the physics of device operation. This approach provides predictive capabilities, and information about the conditions inside a device, but it can require significant amounts of CPU time. Information is usually transferred from device simulation to circuit simulation as follows: Electrical characteristics are calculated using a physically-based device simulator. These calculated electrical characteristics are then used as input by a device modeling and parameter extraction package such as UTMOST. The extracted parameters are used to characterize a compact model used by the circuit simulator.

This approach is adequate for many purposes, but has limitations. It requires that satisfactory compact models already exist. The use of compact models always introduces some error, and models

that are adequate for digital circuit simulation may be inadequate for other applications. Applications and devices for which compact modeling is not always satisfactory include: precision low power, high power, high frequency circuit simulation, SOI, IGBT, GTO, TFT, and optoelectronic devices.

Advantages of MIXEDMODE Simulation

The limitations of compact models can be overcome by using physically-based device simulation to predict the behavior of some of the devices contained in a circuit. The rest of the circuit is modeled using conventional circuit simulation techniques. This approach is referred to as 'mixed-mode' simulation, since some circuit elements are described by compact models, and some by physically-based numerical models.

MIXEDMODE simulation provides several worthwhile advantages. No compact model need be specified for a numerical physically-based device. The approximation errors introduced by compact models can be avoided particularly for large signal transient performance. In addition the user can examine the internal device conditions within a numerical physically-based device at any point during the circuit simulation. The cost is increased CPU time over SPICE, however CPU time is comparable to a device simulation excluding the external circuit nodes. Mixed-mode simulation normally uses numerical simulated devices typically only for critical devices. Non-critical devices are modeled using compact models.

Using MIXEDMODE

Syntax Overview

Input file specification for MIXEDMODE is different in many respects to the rest of ATLAS. However users familiar with SPICE and ATLAS syntax should have little difficulty understanding how these two syntax styles are joined in MIXEDMODE.

Each input file is split in two parts. The first part is SPICE-like and describes the circuit netlist and analysis. The second part is ATLAS-like and describes the device simulation model parameters. These two sections of an input file are separated as described in the next section.

The circuit description includes the circuit topology (called "netlist") and the electrical models and parameters of the circuit components. The simulation conditions specify the types of analysis to be performed. These items are described using syntax based on SPICE.

The ATLAS device descriptions provide information about device geometry, doping distribution, meshes. Device descriptions can be prepared using the built-in ATLAS syntax, the ATHENA process simulator, or the DevEdit structure specification and meshing tool. Previously calculated device solutions may optionally be read in. The device data is read in from standard structure format files.

When a simulation has finished, the following information is available:

- I-V data (voltages in all circuit nodes and currents in all circuit branches)
- Internal distributions of solution variables (such as electron, hole, and potential distributions) within the numerical devices

The results of previous runs of MIXEDMODE can be used as initial guesses for future simulations. This is particularly helpful when multiple simulations must be performed from the same starting point.

The accessing and running of examples for ATLAS are documented in the DECKBUILD chapter of the VWF Interactive Tools manual. It is recommended for all users to run at least one MIXEDMODE example provided on the distribution tape before trying their own simulations.

General Syntax Rules

The SPICE-like part of any MIXEDMODE input file starts with the parameter .BEGIN. The SPICE-like part of the input file ends with .END. All parameters related to the device simulation models appear after the .END statement.

The first non comment statement after initializing ATLAS (go atlas) has to be .BEGIN. The order of the following netlist and control statements is arbitrary, but the last SPICE-like statement has to be .END.

Unlike the rest of ATLAS for the SPICE-like statements the exact command has to be used, unique abbreviations are not accepted. Statements are not case sensitive.

There has to be at least one numerical ATLAS device ("A" device) within the netlist.

Comment characters are "#" and "$", but not "*".

All ATLAS statements specifying the parameters for the numerical device simulation have to be specified after .END

After all ATLAS statements the simulation has to be explicitly terminated (quit, go <simulator>)

These rules do not apply to the SET statement for parameterization of the input file, since it is interpreted by DECKBUILD only.

EXTRACT statements are also an exception similar to SET. Since MIXEDMODE input files are parsed completely before execution (see also "Things to know, Recommendations"), extractions can only be done after completion of the simulation. To extract results from a MIXEDMODE simulation, EXTRACT should be specified after re-initialization of ATLAS (go atlas).

Circuit and Analysis Specification

The SPICE-like MIXEDMODE statements can be divided into three categories:

- element statements defining the circuit netlist,
- simulation control statements specifying the analysis to be performed
- special statements typically related to numerics and output (first character being a dot ".").

The specification of the circuit and analysis part has to be bracketed by a .BEGIN and an .END statements, i.e. all MIXEDMODE statements before .BEGIN or after .END will be ignored or regarded as an error. Between .BEGIN and .END, the order is arbitrary.

Netlist Statements

Each device in the circuit is described by an element statement. The element statement contains the element name, the circuit nodes to which the element is connected and the values of the element parameters. The first letter of an element name specifies the type of element to be simulated. For example, a resistor name must begin with the letter R, and can contain one or more characters. This means that R, R1, RSE, ROUT, and R3AC2ZY are all valid resistor names. Some elements, such as diodes and transistors, must always refer to a model. A set of elements can refer to the same model. For some elements, such as resistors and capacitors, model referencing is optional. Each element type has its own set of parameters. For example, a resistor statement can specify a resistance value after the optional model name. The bipolar transistor (Q) statement can specify an area parameter. All parameters have corresponding default values (see chapter 8, "Compact and User Defined Model Reference"). Independent voltage and current sources have different specifications for transient, DC, and AC phases of simulation. Transient specifications use the keywords EXP, PULSE, GAUSS, SFFM, SIN and TABLE. AC parameters start with the key-word AC.

Elements to be simulated numerically are defined as “A” devices (ATLAS devices). At least one ATLAS device in a circuit is mandatory.

MIXEDMODE supports the use of the following circuit elements:

- Resistors ("R" devices, may be time dependent)
- Capacitors ("C" devices)
- Inductors ("L" devices)
- Independent voltage sources ("V" devices, may be time dependent)
- Independent current sources ("I" devices, may be time dependent)
- Dependent voltage and current sources ("E", "F", "G", "H" devices)
- Coupled (mutual) inductors ("K" devices)
- Lossless transmission lines ("T" devices)
- Diodes ("D" devices)
- Bipolar junction transistors ("Q" devices, Gummel-Poon SPICE model)
- MOSFETs ("M" devices, Level 3 SPICE models)
- User-defined two-terminal elements ("B" devices)
- Optical sources ("O" devices
- Numerically simulated ATLAS devices ("A" devices)

The physical models for linear elements (resistors, capacitors, sources, etc.) are very simple. The models for diodes, BJTs and MOSFETs are described in this chapter. More extensive descriptions of the diode, BJT and MOSFET electrical models can be found in SPICE manuals (such as SmartSPICE).

A node is a point in the circuit where two or more elements are connected. The node numbering is arbitrary with the exception of 0. Node 0 is grounded, i.e., all voltages at nodes are calculated with respect to the voltage at node 0.

As an example, the netlist for the circuit shown in Figure 10-1 is represented by the following MIXEDMODE input deck fragment.

```
# independent voltage source, 0.1V, connected to node 0 (GND) and 1:
V0    1           0           0.1
# 1kOhm resistor, connected to node 1 and 2
R1    1           2           1K
# ATLAS device, connected to node 2 (anode) and 0 (cathode),
# current scaled by 5e7, mesh from file dio.str
ADIO  2=anode     0=cathode  WIDTH=5e7   INFILE=dio.str
```

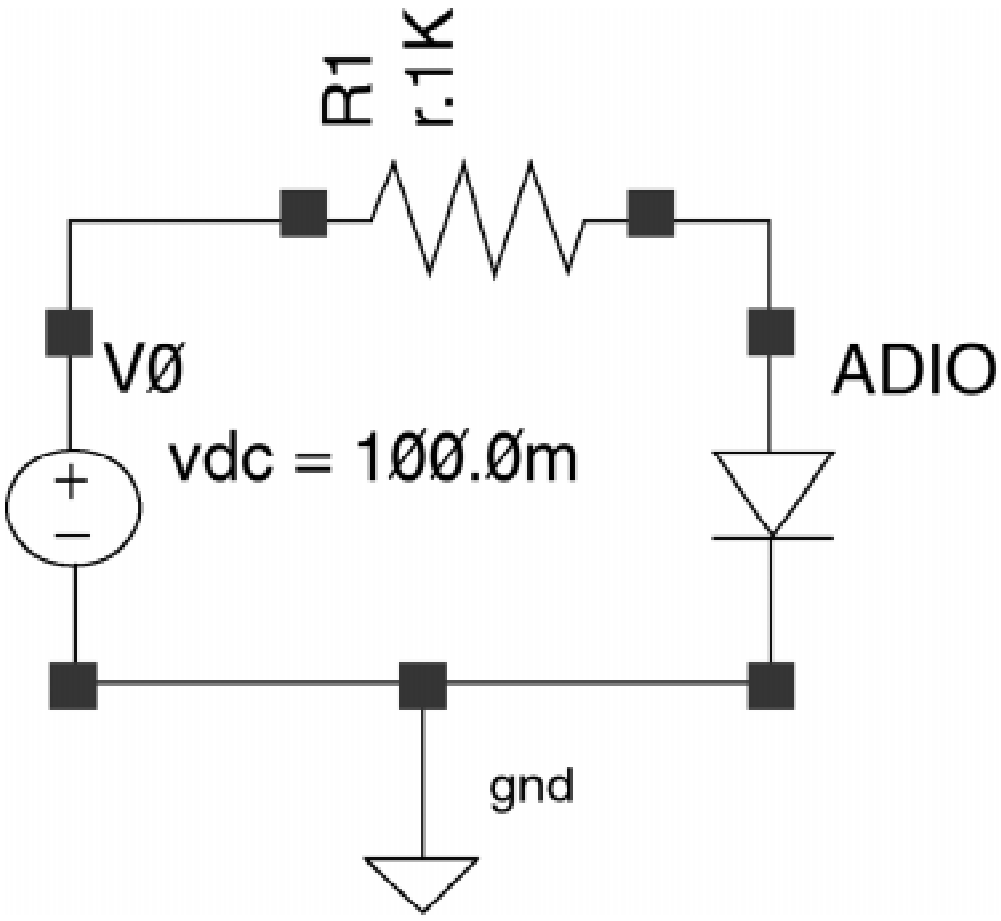

Figure 10-1: Schematic of Primitive Example Circuit

Control Statements

Control Statements are used to specify the analysis to be performed in MIXEDMODE. These take the place of the SOLVE statements in a regular ATLAS input file. At least one of these statements must appear in each MIXEDMODE input file.

- steady state analysis including loops .DC
- transient analysis .TRAN
- small signal AC analysis . AC
- small signal parameter extraction (e.g. s-parameters) .NET

Special statements

Other statements than the control statements beginning with a dot "." specify special parameters for the circuit simulation. These include numerical options and file input and output. Full descriptions of each statement and associated parameters are found later in this chapter.

- compact device models (.MODEL)
- the output files (.LOG, .SAVE)
- initial conditions settings (.NODESET, .IC)

- initial conditions from a file (.LOAD)
- numerics (.NUMERIC, .OPTIONS)
- miscellaneous (.OPTIONS)

Device Simulation Syntax

The second part of a MIXEDMODE command file (after .END) is used to define physical models, material parameters, and numerical methods for ATLAS devices referenced in the "A"-element statements. The following statements may appear in this part of the command file: BEAM, CONTACT, DEFECT, IMPACT, INTERFACE, INTTRAP, MATERIAL, MOBILITY, METHOD, MODELS, OUTPUT, PROBE, TRAP and THERMCONTACT.

It is always necessary to include an indicator to the circuit element name in each device simulation statement even if there is only one A-device in the circuit. All statements specifying the device properties and models are just supplemented by the parameter DEVICE=<name> where <name> is the circuit element in the netlist. <name> will always begin with the letter "A". This makes it possible to define different material properties and model settings for different devices within the circuit.

It is also recommended to specify the REGION parameter referring to only one region in IMPACT, MATERIAL, and MODELS statements. If the device consists of more than one region, several statements with the same device parameters and different region parameters are recommended.

For example to specify the bipolar set of models to a device the syntax used might be:

```
MODEL DEVICE=AGTO REGION=2 BIPOLAR PRINT
```

Recommendations

Input Parsing

In regular ATLAS (non-MIXEDMODE) simulations, the input is interpreted line by line and each statement is executed immediately. This is very useful and nicely supported by DeckBuild for the interactive development of the input. Circuit simulations, however, require the complete input before any simulation can be performed. As a consequence,

a) the complete input is read and parsed before any simulation is initiated,

b) an explicit termination of a simulation is required (quit),

c) all post processing (extraction and plotting) has to be done after re-initializing ATLAS again.

No simulation is started until either a QUIT statement or a GO statement is seen in the input file. Post-processing can be done by re-starting ATLAS.

Scale and Suffixes

In the MIXEDMODE part of the input, numerical values of parameters are represented in standard floating-point notation. The scale suffix may be followed by a unit suffix (e.g., A for Ampere, V for Volt, etc.). Use of a unit suffix can increase the clarity of a command file. The unit suffix is ignored by the program.

The scale suffixes are:

Factor	Name	Suffix
10^{-15}	femto-	F
10^{-12}	pico-	P
10^{-9}	nano-	N

Factor	Name	Suffix
10^{-6}	micro-	U
10^{-3}	milli-	M
10^{3}	kilo-	K
10^{6}	mega-	MG
10^{9}	giga-	G
10^{12}	tera-	T

Numerics

MIXEDMODE solves circuit and device equations simultaneously using fully coupled algorithms. This provides better convergence and requires less CPU time than alternative approaches. The number of circuit variables is often small in comparison with the number of device variables. In this case the CPU time required for simulation performed using MIXEDMODE does not increase drastically compared to the sum of the simulation times required for the individual numerical physically based devices.

MIXEDMODE uses the Newton algorithm for each bias point during steady-state analysis and for each time step during transient analysis. Different variants of the Newton algorithm are used depending on the circumstances. The full Newton method [.OPTIONS FULLN] and a modified two-level Newton method [.OPTIONS .M2LN] are available for steady-state simulation. The full Newton method provides rapid convergence when a good initial guess is available. The modified two-level Newton algorithm is less sensitive to the initial guess. For transient simulation a good initial guess always exists. The full Newton method therefore works very well, and is therefore always used for transient simulation.

When using MIXEDMODE3D, it is recommended that either the DIRECT or GMRES solver be specified. These are specified in the ATLAS part of the MIXEDMODE3D input deck on the METHOD statement.

Multi-Device Structure Representation

If more than one ATLAS device is defined in a MIXEDMODE simulation, the structures are merged together internally. The output solution file is a single file which contains both structures. The first structure referenced will be on top, all other structures will be attached below.

Example: A diode and a bipolar transistor are specified as numerical devices with the following element statements:

```
ABJT 1=BASE 2=EMITTER 4=COLLECTOR WIDTH=1E4 INFILE=bjt.str

ADIO 3=ANODE 4=CATHODE WIDTH=1.5E5 INFILE=dio.str
```

After outputting the solution with:

```
.SAVE MASTER=mas
```

The solution file for the first DC-point mas_dc_1 will contain both structures, the second one (diode) shifted downwards (see Figure 10-2).

This coordinate shift has to be accounted for eventually when extracting position dependent solution quantities or when defining spatially dependent properties with the C-Interpreter.

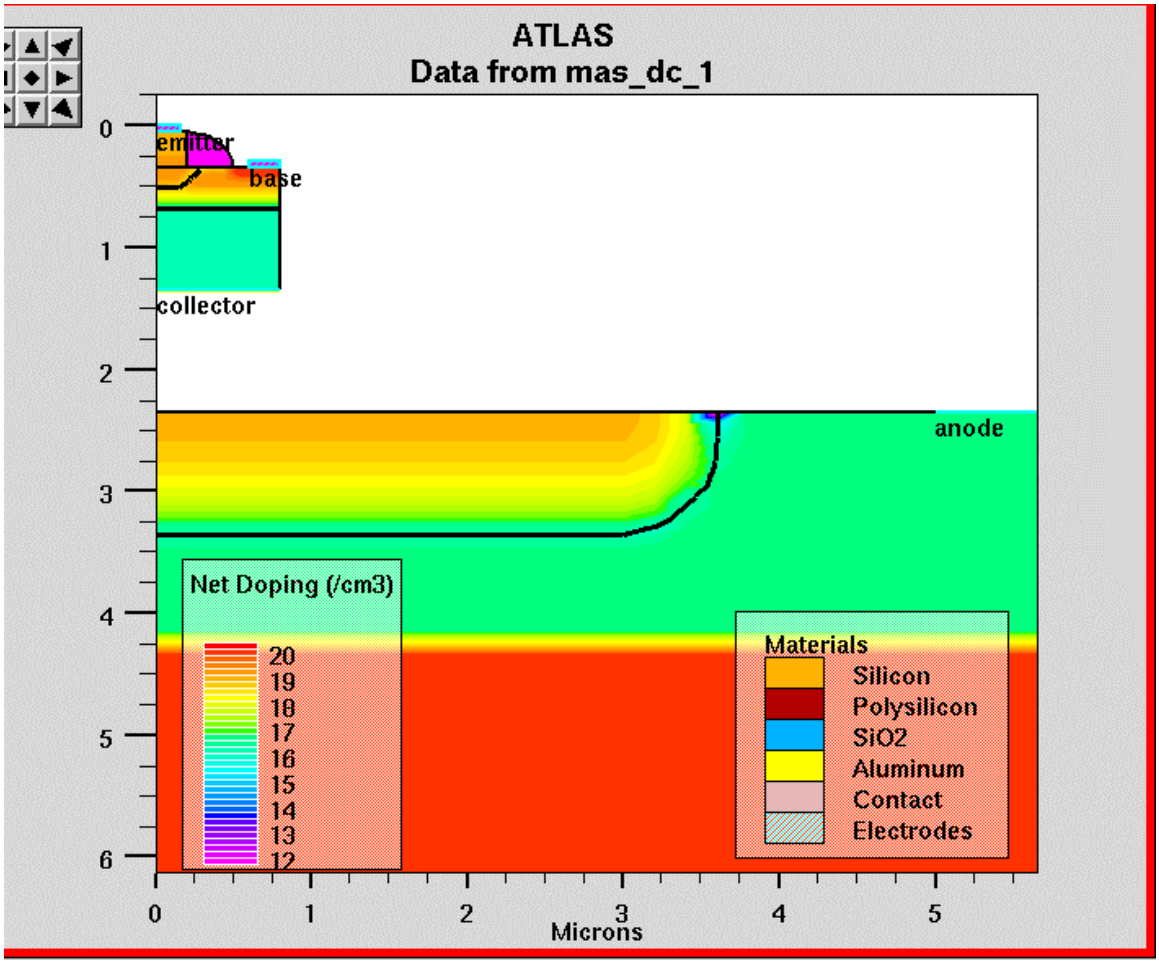

Figure 10-2: Display of a MIXEDMODE solution with two Numerical Devices

Extraction of Results

By default, EXTRACT reads it's data from the currently opened log-file when executed along with ATLAS. Since the extraction of MIXEDMODE log-files require a re-initialization of ATLAS (see "Input Parsing"), EXTRACT has to be initialized explicitly with the correct name of the MIXEDMODE log file. To extract voltages at specific nodes, the syntax `vcct.node."<circuit node>"` has to be used, for extraction of circuit elements `icct.node."<circuit element>"`.

Example:

specify log file:

```
.LOG OUTFILE=hallo
```

Subsequent extraction from the transient log-file is done with:

```
go atlas
extract init inf="hallo_tr.log"
extract name="t0" x.val from curve(time,icct.node."Adio_anode") \
        where y.val=0
```

It extracts the time "t0", when the transient of the current from the "anode" electrode of the device "adio" in the circuit crosses zero. For more details for the EXTRACT syntax refer to the "Interactive Tools" manual.

Using MIXEDMODE inside the VWF Automation Tools

Like all other **Silvaco** products, MIXEDMODE is fully integrated into the VWF framework and can be used for automated experiments. There are some points, however, to take into account:

a) The auto-interface feature does not work with MIXEDMODE. All structures have to be explicitly saved in unique files previous to the MIXEDMODE runs and referred to in the "A"-element statements.

b) MIXEDMODE is not re-entrant, which means that splits within MIXEDMODE runs are not possible. To overcome this problem, the SET statement should be used to define a variable in a process simulator or in the dummy "internal" run. This variable is used to parameterize the input file.

Example: Capacitance as an independent split variable in a VWF experiment:

```
go internal
# define the independent split variable in a re-entrant simulator:
set cap=5e-9

go atlas

.BEGIN
# use the variable as parameter in MIXEDMODE:
C1    2     3     $cap
```

c) The automation tools only store files opened by the normal ATLAS LOG statement in the VWF database, but ignore those defined by .LOG. To overcome this re-initialize ATLAS, open the relevant log file of the previous MIXEDMODE run again with log with the append option in order not to reset it.

Example:

MIXEDMODE log-file definition:

```
.LOG OUTFILE=hallo
```

Re-opening the second DC-log file and the transient log file to get them stored in the VWF database:

```
go atlas

log outfile=hallo_dc_2.log append
log outfile=hallo_tr.log append
```

Initial Settings

Initial convergence is critically dependent on the initial settings of the node voltages (.IC,.NODESET). There should not be any problem starting from the zero bias case. Similarly starting from a preceding MIXEDMODE solution is simple, since the complete solution of the circuit and the ATLAS devices is directly available (.LOAD, .SAVE).

However, when loading solutions for the numerical devices from ATLAS using .OPTIONS LOADSOLUTIONS, sometimes precise matching of the initial circuit condition is required. In this case it is practical to extract the relevant properties in the preceding ATLAS run and use them to parameterize the MIXEDMODE input.

In the following example, the voltages and current of an ATLAS solution is extracted, and the results are used for the initial definition of the circuit.

End of the first part, the stand-alone ATLAS simulation:

```
# extract the final voltage drop on the anode:
extract name="Von" max(vint."anode")
# extract the gate current:
extract name="I_gate" y.val from curve(vint."anode",i."gate") \
      where x.val = $"Von"
extract name="V_gate" y.val from curve(vint."anode",vint."gate") \
      where x.val = $"Von"

# now the MIXEDMODE part

go atlas
.BEGIN
# define the gate current source, use extracted value as parameter
I1 0 7 $"I_gate"
#
# use extracted gate bias and other expressions to calculate
# the node settings:
set Rgl = 10.5
set v7= $V_gate + $I_gate * $Rgl
.NODESET V(1)=2000 V(2)=$"Von" V(3)=$"V_gate" V(4)=$"V_gate" V(5)=-25 \
      V(6)=-15 V(7)=$"v7"
```

A Sample Command File

A sample MIXEDMODE command file is shown below. This file was used to simulate the reverse recovery of a power diode. Several MIXEDMODE examples are provided with the product which can be accessed using DECKBUILD.

```
1. go atlas
2. .BEGIN
3. V1 1 0 1000.
4. R1 1 2 1m
5. L1 2 3 2nH
6. R2 4 0 1MG EXP 1MG 1E-3 0. 20NS 10 200
7. IL 0 4 300
8. ADIODE 3=cathode 4=anode WIDTH=5.E7 INFILE=pd.str
9. .NUMERIC LTE=0.3 TOLTR=1.E-5 VCHANGE=10.
10. .OPTIONS PRINT RELPOT WRITE=10
11. $
12. .LOAD INFILE=pdsave
13. .LOG OUTFILE=pd
14. .SAVE MASTER=pd
15. $
16. .TRAN 0.1NS 2US
17. $
18. .END
19. $
20. MODELS   DEVICE=ADIODE REG=1 CONMOB FLDMOB CONSRH AUGER BGN
21. MATERIAL DEVICE=ADIODE REG=1 TAUN0=5E-6 TAUP=2E-6
```

```
22. IMPACT   DEVICE=ADIODE REG=1 SELB<
23. $
24. METHOD   CLIM.DD=1.E8 DVMAX=1.E6
25. $
26. go atlas
27. TONYPLOT pd_tr.log
```

Line by Line Description

Line 1: All ATLAS input files should begin with `go atlas`

Line 2: The `.BEGIN` and `.END` statements indicate the beginning and end of the circuit simulation syntax. These commands are similar to those used in SPICE.

Lines 3- 7: Circuit components, topology, and analysis are defined within. In general, the circuit component definition consists of three parts; the type of component, the lead or terminal mode assignments, and the component value or model name. For example, the first component definition in this simulation is a DC voltage source. **V1** defines the component as voltage source number one, **1** and **0** are the two circuit modes for this component, and **1000** indicates that the voltage source value is 1000 volts. The remaining circuit components are resistors **R1 R2**, inductor **L1** and independent current source **IL**.

The reverse recovery of the diode is simulated by dropping the value of output resistor **R2** over a small increment of time. The **R2** statement contains additional syntax to perform this task. Here, the resistor is treated as a source whose resistance decreases exponentially from 1 MOhm to 1 mOhm over the specified time step. This action essentially shorts out the parallel current source i1 which is also connected to the base of the diode

Line 8: The `ADIODE` statement specifies a device to be analyzed by ATLAS. The A part of the `ADIODE` command specifies that this is a device statement. The `DIODE` portion simply defines the device name. The option `INFILE=` indicates which device structure file is to be used.

Lines 9-10: These set numerical options for the circuit simulation. `WRITE=10` specifies that every tenth timestep will be saved into the solution file specified on the .SAVE statement.

Line 12: specifies a file generated by a previous `MIXEDMODE` simulation to be used as an initial guess to the voltage.

Line 13-14: specifies the output log and solution filenames. These names are root names and extensions will be added automatically by the program.

Line 16: indicates the type of analysis required. In this case it is a transient simulation lasting 2 microseconds with an initial timestep of 0.1 nanoseconds.

Line 18: indicates the end of the circuit description. All following statements will be related to the ATLAS device.

Lines 20-22: To completely specify the simulation, the physical models used by ATLAS must be identified. Note that `DEVICE=ADIODE` must be specified for each line. The `MODEL` statement is used to turn on the appropriate transport models. This set includes:

- `conmob`: the concentration dependent mobility mode.,
- `fldmob`: the lateral electric field-dependent mobility model,
- `consrh`: Shockley-Read-Hall recombination using concentration dependent lifetimes,
- `auger`: recombination accounting for high level injection effects,
- `bgn`: band gap narrowing.

The MATERIAL statement is used to override default material parameters. In this case, the carrier recombination fixed lifetimes are set. Finally, the Selberherr impact ionization model is enabled using the IMPACT statement with the SELB option.

Line 24: The METHOD statement specifies numerical options for the device simulation. The METHOD statement must come after all other device simulation statements.

Line 26: The command GO ATLAS or a QUIT statement is needed to initiate simulation. Since a plot of the final log file is desired the GO ATLAS option is used to restart ATLAS after the end of the MIXEDMODE simulation.

Line 27: The TONYPLOT command is used to plot the resulting log file.

MIXEDMODE Syntax

This section is split into two parts:

- circuit element statements to describe the netlist
- control and analysis statements

Circuit Element Statements

A – ATLAS device to be simulated using device simulation

Syntax

```
Axxx n1=name1 n2=name2 [n3=name3 ...] infile=filename [width=val]
```

Description

This statement defines a device to be represented by a numerical ATLAS model. The device description with all necessary information (geometry, mesh, doping, models, electrode names, etc.) must be available in a standard structure file prior to starting a MIXEDMODE simulation.

Axxx specifies the name of the element. It must begin with an "**A**".

n1, name1 specifies the circuit node number to which the ATLAS device electrode with the name "name1" is connected. The ATLAS device must have at least two electrodes.

The maximum number of electrodes allowed in ATLAS is 50. This means that up to 25 ATLAS devices can be specified (for example, 25 devices with 2 electrodes each, or 10 devices with 5 electrodes each may be specified). The use of ATLAS device models should be sparing, since it can be very time consuming. Use circuit models for less important circuit components to conserve CPU time.

infile specifies the name of standard structure format file with device geometry, mesh, doping, electrodes names, etc. The number of electrodes and their names should match those mentioned in this statement.

Optionally, this file can contain a solution, which MIXEDMODE will use as an initial guess (see the .OPTIONS statement for more details).

width is an optional parameter (default = 1). All currents through ATLAS device terminals calculated using the 2-D ATLAS model will be multiplied by this parameter to account for the third dimension of the device. **Width** can still be used as a multiplier to the ATLAS current if a 3D ATLAS structure is used in MIXEDMODE3D.

Example

```
ABJT1 3=EMITTER 4=BASE 6=COLLECTOR INFILE=BJT1.STR WIDTH=10
```

B – User-defined two-terminal element

Syntax

```
Bxxx n+ n- INFILE=<file_name> FUNCTION=<function_name>
```

where:

Bxxx specifies the element name of a user-defined two terminal device. It must begin with a "**B**".

n+, n- are the positive and negative terminal node numbers.

file_name specifies the name of the text file that contains C source code for a user-defined function that describes element behavior. This file can contain more than one function description.

function_name specifies the name of the function from the file.

Example

```
B1 2 3 infile=ud.c function=rc
```

Note: More detailed documentation on the user-defined function is at the end of this chapter

C – Capacitor

Syntax

Cxxx n+ n- value

Description

Cxxx specifies the name of a capacitor element. It must begin with a "**C**".

n+, n- are the positive and negative terminal node numbers.

value is the capacitance in farads.

Example

Cload 3 0 1pF

D – Diode

Syntax

Dxxx n+ n- mname [area]

Description

Dxxx specifies the name of the diode element. It must begin with a "D".

n+, n- are the positive and negative terminal node numbers.

mname is the diode model name. It must refer to a diode model.

area is the area factor. The default is 1.0.

Example

D1 2 3 dmodel1 2.e3.

E – Linear voltage controlled source

Syntax

```
Exxx n+ n- nc+ nc- gain
```

Description

Exxx specifies the name of the linear voltage controlled voltage source. It must begin with an "E".

n+, n- are the positive and negative terminal node numbers. A positive current flows from the node n+, through the source, to the node n-.

nc+, nc- are the positive and negative controlling node numbers.

gain is the voltage gain.

The linear voltage-controlled voltage source is characterized by the equation:

v(v+, n) = gain * v(nc+,nc)

Example

ER 4 5 6 7 55.

F– Linear current controlled current source

Syntax

```
Fxxx n+ n- vcontrolname gain
```

Description

Fxxx specifies the name of the linear current controlled current source. It must begin with an "F".

n+, n- are the positive and negative terminal node numbers. A positive current flows from the node n+, through the source, to the node n-.

vcontrolname specifies the name of the voltage source through which the controlling current flows. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of vcontrolname.

gain specifies the current gain.

The linear current-controlled current source is characterized by the equation:

i(n+, n-) = gain * i(vcontrolname).

Example

F12 4 5 VIN 0.1

G – Linear voltage controlled current source

Syntax

```
Gxxx n+ n- nc+ nc- trcon
```

Description

Gxxx specifies the name of the linear voltage controlled current source. It must begin with a "G".

n+, n- are the positive and negative terminal node numbers. A positive current flows from the node n+, through the source, to the node n-.

nc+, nc- are the positive and negative controlling node numbers.

trcon is the transconductance (in 1/Ohms).

The linear voltage controlled current source is characterized by the equation:

i(n+, n-) = trcon * v(nc+,nc-).

Example

G2 4 5 6 7 5.

H – Linear current controlled voltage source

Syntax

```
Hxxx n+ n- vcontrolname trres
```

Description

Hxxx specifies the name of the linear current controlled voltage source. Must begin from an "H".

n+, n- are the positive and negative terminal node numbers. A positive current flows from the node n+, through the source, to the node n-.

vcontrolname is the name of voltage source through which the controlling current flows. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of vcontrolname.

trres is the transresistance (in Ohms).

The linear current controlled voltage source is characterized by the equation:

v(n+, n-) = trres * i(vcontrolname)

Example

H12 4 5 V1 0.1K

I – Independent current source

Syntax

```
Ixxx n+ n- value [ac_parameters] [transient_parameters]
```

Description

Ixxx specifies the name of the independent current source. It must begin with an <@147>I<@148>.

n+, n- are the positive and negative terminal node numbers.

ac_parameters are described on page 10-23.

value is the DC value of the source (ampers).

transient_parameters are described on page 10-24.

Example

I1 2 8 0. PULSE 0 200 0 20ns 20ns 100ns 10 100

K – Coupling between two inductors

Syntax

Kxxx Lyyy Lzzz kval

Description

This is not a real circuit element. This statement defines only the coupling between two inductors.

Kxxx specifies a name. This parameter is not important and is used only to distinguish the statement. It must begin with a "K".

Lyyy specifies the first inductor element name. It must begin with an "L" and match one of the inductor names from the circuit.

Lzzz specifies the second inductor element name. It must begin with an "L" and match one of the inductor names from the circuit.

kval is the coefficient of mutual coupling, which must be in the range 0 < kval < 1. The mutual inductance M will be determined from the relation:

$$M = kval * L1 * L2$$

Example

K1 L22 LLOAD 0.99

L – Inductor.Syntax

Lxxx n+ n- value

Description

Lxxx specifies the inductor name. It must begin with an "L".

n+, n- are the positive and negative terminal node numbers.

value is the inductance in Henrys.

Example

L2 2 3 2.5nH

M – MOSFET

Syntax

Mxxx nd ng ns nb mname [L=val] [W=val] [AD=val] [AS=val]

[PD=val] [PS=val] [NRD=val] [NRS=val]

Description

Mxxx specifies the MOSFET element name. It must begin with an "M".

nd, ng, ns, nb are the drain, gate, source, and bulk terminal nodes numbers, respectively.

mname is the model name. It must refer to a MOSFET model.

AD specifies the area of the drain.

AS specifies the area of the source.

NRD specifies the number of squares for drain resistance.

NRS specifies the number of squares for source resistance.

PD specifies the periphery length of the drain.

PS specifies the periphery length of the source.

Example

M1 2 3 4 5 mos1 L=1u W=30u

O – Optical source

Syntax

Oxxx beam value [transient_parameters]

Description

Oxxx specifies the name of an independent optical source. It must begin with an "O".

beam specifies the beam number. The beam with this number should be described in the ATLAS section of the command file. See the LUMINOUS chapter for a complete description of optoelectronic simulation.

value is the DC optical intensity value (W/cm2).

transient_parameters are described later in this chapter.

Note: The treatment of optical sources is fully similar to the treatment of independent voltage/current sources, i.e, .DC statements can be used to simulated DC light responses of the circuit and transient parameters can be used to describe the transient behavior of the optical sources.

Example

O1 1 0.001 pulse 0.001 0.002 0 2ns 2ns 100ns 10 100

Q – Bipolar junction transistor

Syntax

```
Qxxx nc nb ne ns mname [area]
```

Description

Qxxx specifies the name of a bipolar junction transistor. It must begin with a "Q".

nc, nb, ne, ns are the collector, base, emitter, and substrate node numbers.

mname is the model name. It must refer to a BJT model.

area is the area factor. The default value is 1.0. This parameter is optional.

Example

```
Qinv 2 5 7 0 bjtmod1 20.
```

R – Resistor

Syntax

```
Rxxx n+ n- value [transient_parameters]
```

Description

Rxxx specifies the name of the resistor element. It must begin with an "R".

n+, n- are the positive and negative terminal node numbers.

value is the resistance in ohms.

transient_parameters are described on page 10-22.

Note: Unlike the traditional **SPICE** program, transient parameters are acceptable for resistor elements. This allows simulation of different kinds of time-dependent resistors and switches in a very simple way.

Example

```
R12 4 5 100k
```

T – Lossless transmission line

Syntax

```
Txxx n1 n2 n3 n4 Z0= val TD=val
```

Description

Txxx specifies the name of the transmission line element. It must begin with a "T".

n1, n2 are the nodes at port 1.

n3, n4 are the nodes at port 2.

Z0 is the characteristic impedance.

TD is the transmission delay.

Example

T1 1 0 2 0 Z0=50 TD=10ns

V – Independent voltage source

Syntax

```
Vxxx n+ n- value [ac_parameters] [transient_parameters]
```

Description

Vxxx specifies the name of the independent voltage source. It must begin with a "V".

n+, n- are the positive and negative terminal node numbers.

ac_parameters are described on page 10-21

value specifies the DC value of the source in units of volts.

transient_parameters are described on page 10-22.

Example

VCC 5 0 10.

Control and Analysis Statements

.BEGIN

.BEGIN - indicates the start of the circuit part of a `MIXEDMODE` command file.

.END

.END - indicates the end of the circuit part of a `MIXEDMODE` command file.

.AC

.AC - performs an AC linear small-signal analysis on the circuit. `MIXEDMODE` first creates a linearized small-signal model at the operating point of the circuit, and the computes the frequency response over a user-specified range of frequencies.

Syntax

```
.AC DEC|OCT|LIN nump fstart fstop.
```

Description

DEC specifies that the frequency sweep is by decades.

OCT specifies that the frequency sweep is by octaves.

LIN specifies that the frequency sweep is linear. This is default.

nump is the total number of points per decade or per octave, or the total number of points of the linear sweep.

fstart is the starting frequency (Hz).

fstop is the final frequency (Hz).

Several .AC statements can be specified in the same command file. In this case, they will be executed sequentially. Before the execution of the first .AC statement, the program will execute all .DC statements (if any), regardless of the order of the .AC and .DC statements in the command file.

Examples

.AC DEC 3 1.e3 1.e12

.AC LIN 20 1.e5 2.e6

.DC

.DC - causes a DC transfer curve to be computed for the circuit with all capacitors opened and all inductors shorted.

Syntax

```
.DC     DEC|OCT|LIN source_name start stop numbers_steps
       source_name2 DEC|OCT|LIN start2 stop2 number_steps2
```

Description

source_name is the name of the independent voltage or current source to be swept.

source_name2 is the name of the secondary sweep source.

start is the starting value of the sweep argument.

start2 is the start value of the secondary sweep source

stop is the final value of the sweep argument.

stop2 is the final value of the secondary sweep source

number_steps is the number of steps of the inner sweep

number_steps2 is the number of steps of the secondary sweep.

DEC is the DC bias (voltage or current) sweep by decades.

OCT is the DC bias sweep by octaves.

LIN is a linear DC bias sweep. This is the default.

Several .DC statements can be specified in a command file. In this case they will be executed sequentially. Before executing the first .DC statement, the program will simulate the circuit with the independent source values given in the description of those sources.

The .DC statement is also often used to increment the values of independent voltage and current sources in a circuit to avoid convergence problems.

Examples

```
.DC VIN 0. 5. 0.25
.DC IE 50 500 50
```

.NET

.NET - specifies that a network parameter extraction is to be performed.

Syntax

```
.NET inport outport DEC|OCT|LIN nump fstart fstop [Z0] [INDIN] [RSIN] [INDOUT]
[RSOUT] [CIN] [COUT].
```

Description

inport is the input port description. It should be in one of the following formats.

V(n+,n-) – two nodes (positive (n+) and negative (n-)).

Vxxxx – where Vxxxx is the name of an existing voltage source. The positive terminal of the source becomes the positive input port node and the negative terminal becomes the negative input node.

Ixxxx – where **Ixxxx** is the name of an existing current source. The positive terminal of the source becomes the positive input port node and the negative terminal becomes the negative input node.

Note: If the nodes specified as the input port are the same nodes as an existing current or voltage source, then the name of the source MUST be specified as inport.

---

outport is the output port description. It should be in one of the following formats.

V(n+,n-) – two nodes (positive (n+) and negative (n-)).

Vxxxx – where **Vxxxx** is the name of an existing voltage source. The positive terminal of the source becomes the positive output port node and the negative terminal becomes the negative output node.

Ixxxx – where Ixxxx is the name of an existing current source. The positive terminal of the source becomes the positive output port node and the negative terminal becomes the negative output node.

Note: If the nodes specified as the output port are the same nodes as an existing current or voltage source, then the name of the source MUST be specified as outport. Also, all AC parameters should be removed from voltage or current sources before using the .NET statement.

DEC specifies that the frequency sweep is by decades.

OCT specifies that the frequency sweep is by octaves.

LIN specifies that the frequency sweep is linear. This is default.

nump is the total number of points per decade or per octave, or the total number of points of the linear sweep.

fstart is the starting frequency (Hz).

fstop is the final frequency (Hz).

Additional optional parameters may also be specified on the .NET statement.

Z0 specifies the matching Impedance (default = 50 Ohms).

INDIN specifies the inductance through which the DC voltage source is connected to the input source (only if INPORT is given as Vxxxx).

RSIN specifies the series resistance of INDIN.

INDOUT specifies the inductance through which the DC voltage source is connected to the output source (only if OUTPORT is given as Vxxxx).

RSOUT specifies the series resistance of INDOUT.

CIN specifies the capacitance through which the S-parameter test circuit is connected to the input port.

COUT specifies the capacitance through which the S-parameter test circuit is connected to the output port.

Note: The S-parameters will be automatically saved to the LOG file. The Z, Y, H, ABCD and gain small-signal parameters can also be written to the LOG file. These are selected via the .OPTIONS statement. Also, the default values can be viewed if PRINT is specified on the .OPTIONS statement

---

Examples

```
.NET V1 V2 DEC 10 1e6 1e10
.NET I1 V2 DEC 10 1e6 1e10 Z0=75 RSOUT=100
.NET V(1,0) V(2,3) DEC 10 1e6 1e10
```

.TRAN

.TRAN - specifies that a transient analysis is to be performed.

Syntax

```
.TRAN tin tstop.
```

Parameter	Type	Default	Units
tin	Real		s
tstop	Real		s

Description

tin is the initial timestep.

tstop is the final time value for which the simulation is to be performed.

Transient analysis is performed only after the execution of all .DC statements. If no DC statements are used the transient starts after the calculation of the initial circuit state with the values of the independent sources given in the descriptions of those sources.

Multiple .TRAN statements are supported. The TSTOP parameter is NOT reset between each .TRAN statement.

Example

```
.TRAN 1ns 100ns
```

.LOAD

.LOAD - loads a solution file.

Syntax

```
.LOAD INFILE=<filename>
```

Description

INFILE specifies the name of a file to be loaded as an initial guess for further simulation. This file must have been saved during a previous run of MIXEDMODE using the .SAVE statement.

Example

```
.LOAD INFILE=pdsave
```

Note: This statement is *not* used to load SSF format solution files from ATLAS (see .OPTIONS LOADSOLUTIONS)

.SAVE

.SAVE saves simulation results into files for visualization or for future use as an initial guess.

Syntax

```
.SAVE [outfile=name] [master=mname]
```

Description

outfile specifies that after the simulation is finished the solution is to be written to a file called **name**. The ATLAS model solutions will be written to the file **name** and the circuit solution will be written to the file **name.cir**. These files can be used later for loading solutions to be used as an initial guess (see .LOAD statement).

master specifies that the internal states of all ATLAS models should be written during the simulation in standard structure format for future visualization using TONYPLOT. These files, with the base name **mname**, will be written after the calculation of each bias point during DC simulation, and after of each time step during transient simulation.

The program will automatically add the following suffixes to **mname**:

- During DC simulation: **_dc_number**, where **number** is the number of the DC point.
- During transient simulation: **_tr_numbe**r, where **number** is the number of the time step.

Example

.SAVE OUTFILE=pdsave MASTER=pd

.LOG

.LOG specifies the filename in which circuit voltages and currents will be saved

Syntax

```
.LOG outfile=<filename>
```

Description

outfile specifies the filename in which circuit voltages and currents are to be saved in standard structure format files.

These files will have the following names:

For steady-state analysis:

"filename"_dc_1.log

"filename"_dc_2.log

"filename"_dc_3.log

..

(new file will be created for each .DC statement).

For AC analysis:

"filename"_ac_1.log

..

For network parameter extraction:

"filename"_net_1.log

..

For transient analysis:

"filename"_tr.log

..

To plot results of and entire steady-state analysis simultaneously, load all files related to steady-state analysis into TONYPLOT.

Example

.LOG OUTFILE=pd

.IC

.IC sets specified node voltages during steady-state simulation.

Syntax

```
.IC [V(I)=VAL_I ...]
```

Description

This statement forces the specified node voltages to be set to specified values during steady-state simulation. These voltages are released when transient simulation begins.

Example

.IC V(1)=10.

.NODESET

.NODESET sets initial values for circuit node voltages.

Syntax

.NODESET [V(I)=VAL_I ...]

Description

This statement specifies the initial values for circuit node voltages. If a node voltage is not specified, the program will try to find a solution using 0 as an initial guess for this node. This statement can significantly reduce the CPU time needed to calculate the initial condition.

Example

.NODESET V(1)=50. V(2)=49.4 V(3)=10. V(5)=-1.

.NUMERIC

.NUMERIC specifies special numeric parameters for the circuit analysis.

Syntax

```
.NUMERIC   [<<parameters>>]
```

Parameter	Type	Default	Units
IMAXDC	Integer	25	
IMAXTR	Integer	15	
TOLDC	Real	$1*10^{-4}$	
TOLTR	Real	$1*10^{-4}$	
LTE	Real	0.1	
VMAX	Real	$5*10^{7}$	V

Parameter	Type	Default	Units
VMIN	Real	$-5*10^{7}$	V
VCHANGE	Real	$5*10^{7}$	V
DTMIN	Real	$1*10^{-12}$	s

Description

IMAXDC specifies the maximum number of mixed circuit-device iterations to be performed during steady-state analysis.

IMAXTR specifies the maximum number of mixed circuit-device iterations to be performed during transient analysis.

TOLDC specifies the relative accuracy to be achieved during steady-state analysis for the calculation of voltages in circuit nodes.

TOLTR specifies the relative accuracy to be achieved during transient analysis for the calculation of voltages in circuit nodes.

LTE specifies the local truncation error for transient analysis.

VMAX specifies the maximum value for circuit node voltages.

VMIN specifies the minimum value for circuit node voltages.

VCHANGE specifies the maximum allowable change in circuit node voltages between two mixed circuit-device iterations. (This parameter can be useful for reaching steady-state convergence with a bad initial guess.)

DTMIN specifies the minimum time step value for transient analysis.

Example

.NUMERIC LTE=0.05 TOLDC=1.*10-8 DTMIN=1ns

.OPTIONS

.OPTIONS - specifies various circuit simulation options.

Syntax

```
.OPTIONS [TNOM=<value>] [FULLN|M2LN] [PRINT] [RV=value] [RELPOT] [NOSHIFT]
[WRITE=<value>] [CYLINDR] [LOADSOLUTIONS] [CNODE=<value.]
```

Parameter	Type	Default	Units
TNOM	Real	300	K
FULLN	Logical	True	
M2LN	Logical	False	
PRINT	Logical	False	

Parameter	Type	Default	Units
RELPOT	Logical	False	
NOSHIFT	Logical	False	
RV	Real	$1*10^{-4}$	W
CNODE	Real	$1*10^{-16}$	F
WRITE	Integer	1	
LOADSOLUTIONS	Logical	False	
CYLINDER	Logical	False	
Z.PARAM	Logical	False	
Y.PARAM	Logical	False	
H.PARAM	Logical	False	
ABCD.PARAM	Logical	False	
GAINS	Logical	False	

Description

TNOM specifies the circuit temperature to be use during the simulation.

FULLN, **M2LN** specifies the solution method to be used during steady-state simulation.

FULLN specifies the full Newton method.

M2LN specifies the modified two-level Newton method.

The full Newton method provides faster solution than the modified two-level Newton method when a good initial guess is available. The modified two-level method is more reliable when the initial guess is far from the solution. The default is the full Newton method.

PRINT enables printing of circuit nodes voltages after the calculation for each bias point (DC analysis) or time step (transient the analysis).

RELPOT enables the use of relative convergence criteria for potential for ATLAS models. By default, **ALTAS** models use absolute convergence criteria for potential. When bias voltages are large (a common situation for power devices), then absolute convergence criteria are not appropriate and this parameter should be specified.

NOSHIFT disables the shift of voltages for ATLAS device models. MIXEDMODE normally shifts the voltages on ATLAS device terminals to be referenced to the voltage on the first terminal. From the physical point of view, the state of the p-n diode is the same for voltages of 0V and 0.5V on the diode terminals with 1000V and 1000.5V, but the first situation is better for numerical simulation.

RV defines the ohmic resistance that MIXEDMODE associates with all voltage sources and all inductances. This value should never be set to 0. The default value is small enough to avoid errors due to the influence of the internal resistance. Usually, extremely small values of this parameters can cause convergence problems. It is usually acceptable to decrease this parameter to the range of $1*10^{-6}$-$1*10^{-7}$. This parameter should not be varied unless there is a compelling reason to do so.

CNODE specifies a very small capacitance that is for algorithmic reasons automatically connected from each circuit node to ground. This value can be set to 0.

WRITE specifies how often the solution is to be saved in standard structure files during the simulation. For example, write=3 specifies that the solution will be saved at every third timestep. Specifying this parameter can help avoid disk overflow.

CYLINDR activates the cylindrical coordinate system for all ATLAS devices.

LOADSOLUTIONS indicates that solutions, as well as structures, doping distributions, and meshes, are to be loaded from standard structure files. The solutions are used as the initial guess or initial conditions for subsequent MIXEDMODE simulation. To use this feature, the user must:

1. Calculate a solution for each ATLAS device, and save each solution in a separate standard structure format file using SAVE or SOLVE.... MASTER.
2. For each ATLAS device, use the A statement in the MIXEDMODE command file to specify the associated standard structure format file
3. Set node voltages to appropriate values with the .NODESET statement
4. Specify **LOADSOLUTIONS** in the .OPTIONS statement

Note: If using this feature, you must specify solutions for all ATLAS devices.

The .NODESET statement must always be used when LOADSOLUTIONS is used. The .NODESET statement is used to make the initial circuit voltages match those for which device solutions were obtained. It may also be necessary to specify the NOSHIFT parameter of the OPTIONS statement. By default, MIXEDMODE shifts device terminal voltages with respect to the voltage on the first terminal that is specified in the A statement. You must either prepare initial solutions with this terminal grounded, or specify NOSHIFT in the OPTIONS statement.

Z.PARAM specifies that the Z-parameters should be written to the LOG file. This is used in conjunction with the .NET statement.

Y.PARAM specifies that the Y-parameters should be written to the LOG file. This is used in conjunction with the .NET statement.

H.PARAM specifies that the H-parameters should be written to the LOG file. This is used in conjunction with the .NET statement.

ABCD.PARAM specifies that the ABCD-parameters should be written to the LOG file. This is used in conjunction with the .NET statement.

GAIN specifies that the stability factor (K), unilateral power gain (GU), maximum unilateral transducer power gain (GTUmax) and |H21|^2 are written to the LOG file. This is used in conjunction with the .NET statement.

Example

```
.OPTIONS TNOM=293 FULLN
```

.MODEL

.MODEL – specifies the circuit element model to be used for diodes, BJTs, or MOSFETs, and the numerical values of parameters associated with the model.

Syntax

```
.MODEL <<name>> <<type>> [<<parameters>>]
```

Description

name specifies the name of the model. Circuit element definition statements refer to this name to link elements to models.

type specifies the type of the model. This type must be consistent with the type of the circuit elements that uses the model. The type can be one of the following:

D - Diode model

NMOS - n-channel MOSFET model. This corresponds to the SMARTSPICE Level 3 model.

PMOS - p-channel MOSFET model. This corresponds to the SMARTSPICE Level 3 model.

NPN - npn BJT model

PNP - pnp BJT model

Sets of model parameters exist for each model type. These sets of parameters are described in detail in the model descriptions presented later in this chapter.

Example

```
.MODEL MODBJT NPN IS=1.E-17 BF=100 CJE=1F TF=5PS \
CJC=0.3F RB=100 RBM=20
```

Transient Parameters

Overview

MIXEDMODE allows the user to specify transient parameters for voltage sources (Vxxx), current sources (Ixxx) and resistors (Rxxx). These parameters describe the time development behavior of the source.

PULSE

PULSE is used to define a pulse waveform. The waveform is specified as follows:

```
PULSE i1 i2 td tr tf pw per
```

where:

i1 is the initial value.

i2 is the pulsed value.

td is the delay time before the pulse is started.

tr is the rise time of the pulse.

tf is the fall time of the pulse.

pw is the pulse length per period

per is the period.

The transient behavior is described by the following table. Intermediate points are found by linear interpolation.

Time	Value
0	i1
td	i1
td + tr	i2
td + tr + pw	i2
td + tr + pw + tf	i1
td + per	i1
td + per + tr	i2 (second period)

EXP

EXP is used to define an exponential waveform. The waveform is specified as follows:

EXP i1 i2 td1 tau1 td2 tau2

where:

i1 is the initial value.

i2 is the pulsed value.

td1 is the rise delay time.

td2 is the fall delay time.

tau1 is the rise time constant.

tau2 is the fall time constant.

The transient behavior will be:

Time	Value
0 < t < td1	i1
td1 ≤ t < td2	i1+(i2-i1).(1-exp[(-t-td1)/tau1])
td2 ≤ t	i1+(i2-i1).(1-exp[(-t-td1)/tau1]) +(i1-i2).(1-exp[(-t-td2)/tau2])

GAUSS

GAUSS is used to define a Gaussian waveform. The waveform is specified as follows:

GAUSS i1 i2 td1 tau1 td2 tau2

where:

i1 is the initial value.

i2 is the pulsed value.

td1 is the rise delay time.

td2 is the fall delay time.

tau1 is the rise time constant.

tau2 is the fall time constant.

The transient behavior will be:

Time	Value
$0 < t < td1$	$i1$
$td1 \le t < td2$	$i1+(i2-i1).(1-\exp[((-t-td1)/tau1)^2]$
$td2 \le t$	$i1+(i2-i1).(1-\exp[((-t-td1)/tau1)^2]$ $+(i1-i2).(1-\exp[((-t-td2)/tau2)^2]$

SFFM

SFFM is used to define a modulated sinusoidal waveform. The waveform is specified as follows:

```
SFFM io ia fc mdi fs
```

where:

io is the DC offset.

ia is the amplitude.

fc is the carrier frequency.

mdi is the modulation index.

fs is the signal frequency.

The transient behavior will be:

$value(t)=io + ia \cdot \sin[\pi \cdot fc \cdot t + mdi \cdot \sin(2\pi \cdot fs \cdot t)]$

SIN

SIN is used to define a sinusoidal waveform. The waveform is specified as follows:

```
SIN io ia freq td theta
```

where:

io is the offset.

ia is the amplitude.

freq is the frequency.

td is the delay.

theta is the damping factor.

The transient behavior will be:

Time	Value
t < td	value(t) = io
t ≥ td	value(t) = i0+ia.exp[(-t-td)/THETA] .sin[2π.freq.(t-td)]

TABLE

TABLE is used to define a waveform using a table of values. This parameter is used as follows:

```
TABLE infile=<table_file_name>
```

where table_file_name is an ASCII text file that contains the tabulated time-dependence of a variable in the following format:

```
t1 v1
t2 v2
t3 v3
...
tN vN
end
```

Each line contains two numbers. The first number is the time in seconds. The second number is the time-dependent variable) voltage in volts, the current in amps, or the resistance in ohms). Up to 1000 lines can be used. Input is terminated by the word end.

If during the simulation the transient time becomes larger than the last value in the table, then the last value will be used for the remainder of the simulation

AC Parameters

1. MIXEDMODE allows AC parameters to be specified for independent voltage (Vxxx) and current (Ixxx) sources. These parameters describe the AC behavior of the source.

Syntax

AC value

Description

value specifies the AC source value (in V for voltage sources and in A for current sources).

Compact Device Models

This section provides a detailed description of the models and parameters used to describe diodes, bipolar transistors, and MOS transistors. Models are selected using the element and model statements. Parameters associated with model equations are described here, as part of the model description.

Diode Model

The equivalent circuit for the diode model is shown on Figure 10-3.

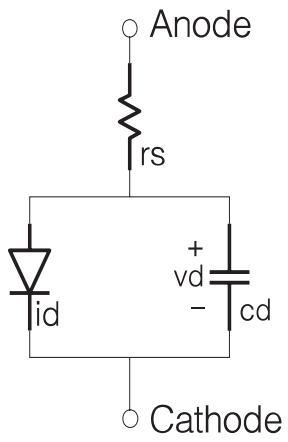

Figure 10-3: Diode Equivalent Circuit

Equations for the diode current (id) equations account for all the fundamental DC effects in the diode. Capacitance effects are assumed to be separate from the id equations, and are modeled independently.

The DC characteristics of the junction diode are determined by the parameters IS, N, and RS. Charge storage effects are modeled by the transit time parameter, TT, and a nonlinear depletion layer capacitance is determined by the parameters, CJO, VJ, and M.

The reverse saturation current IS, emission coefficient N, and ohmic resistance RS are normally obtained from DC measurements of the forward biased diode characteristics. N is determined from the slope of the diode characteristic in the ideal region. In most cases, the emission coefficient has a value of unity.

In practice, at high levels of bias, the diode current deviates from the ideal exponential characteristic. This deviation is due to the presence of ohmic resistance in the diode, and high-level injection effects. The deviation of the actual diode voltage from the ideal exponential characteristic at a specific current determines the value of RS. In practice, RS is estimated at several values of id and averaged, since the value of RS can be dependent on the diode current.

Table 10-1. Diode DC Parameters

Parameter	Description	Units	Default	x Area
IS	Saturation current	A	$1*10^{-14}$ *	
RS	Ohmic resistance	ohm	$1*10^{-4}$	*
N	Emission coefficient		1	

Note: *RS should never be set to 0 in* MIXEDMODE.

Table 10-2. Diode Capacitance Parameters

Parameter	Description	Units	Default	x Area
TT	Transit time	sec	0	
FC	Coefficient for the forward bias depletion capacitance formula		0.5	
CJ0	Zero-bias junction capacitance	F	0	*
M	Junction grading coefficient		0.5	
VJ	Junction potential	V	0.8	

DC Current Equations

This diode model is based on the original Berkeley model.

Define: $v_{te} = N \cdot v_t \qquad \left(v_t = \frac{kT}{q}\right)$ 10-1

For forward bias defined as $v_d > -10 \cdot v_t$

$$i_d = IS_{eff} \cdot \left(e^{\frac{v_d}{v_{te}}} - 1\right) \quad \text{10-2}$$

$$g_d = \frac{IS_{eff}}{v_{te}} \cdot e^{\frac{v_d}{v_{te}}} \quad \text{10-3}$$

In cases of reverse bias node 1 anode is more negative than cathode. The diode is turned off and conducts a small leakage current.

For reverse bias:

$$i_d = -IS_{eff} \quad \text{10-4}$$

Capacitance Equations

The diode capacitance is modeled by the parameter c_d. This capacitance is a combination of diffusion capacitance (c_{diff}) and depletion capacitance (c_{dep}).

$$c_d = c_{diff} + c_{dep} \quad \text{10-5}$$

Diffusion Capacitance Equations

The diffusion capacitance due to the injection of minority carriers is modeled by the transit time parameter TT. In practice, TT is estimated in time domain from time-delay measurements.

$$c_{diff} = TT \cdot \frac{\partial i_d}{\partial v_d} = TT \cdot g_d \qquad 10\text{-}6$$

Depletion Capacitance Equations

The following equations model the depletion capacitance:

For the reverse biased junction defined as: $v_d < FC \cdot VJ$:

$$c_{dep} = \frac{area_{eff} \cdot CJO}{(1-FC)^{1+MJ}} \cdot \left(1 - FC \cdot (1+MJ) + MJ \cdot \frac{v_d}{VJ}\right) \qquad 10\text{-}7$$

For the forward biased junction defined as: $v_d \geq FC \cdot VJ$:

$$c_{dep} = area_{eff} \cdot CJO \cdot \left(1 - \frac{v_d}{VJ}\right)^{-MJ} \qquad 10\text{-}8$$

BJT Model

To model bipolar junction transistors, MIXEDMODE provides a modified Ebers-Moll model and the integral charge control model of Gummel and Poon. The Ebers-Moll model is a physical model and its parameters can be easily measured and extracted. However, the simplicity of the Ebers-Moll model limits its applications to simple digital circuits. The MIXEDMODE model extends the original Ebers-Moll model to include several effects at high bias levels.

The Gummel-Poon model is accurate and complex with approximately 40 parameters. The model parameters all have physical meaning. This makes the model attractive for complicated circuit designs and process characterization. In cases where some parameters are not specified, the model automatically limits itself to a simpler Ebers-Moll model.

The parameter names used in the modified Gummel-Poon model are more easily understood by circuit designers and better reflect physical-processing and circuit-design thinking. The circuit designer does not necessarily always need to use the entire model, but specifying a complete model is highly recommended.

Model Application

The Ebers-Moll model is specified by the parameters IS, BF, and BR. This model is simple, fast, and easy to use for basic applications. Its main application is to quickly provide initial solutions for complex models and circuits. For large circuits, a complete model should be used and all parameters should be specified to achieve maximum accuracy. Depending on how many parameters are not specified, the Gummel-Poon model will decompose itself into a simpler variant of the Ebers-Moll model.

Convergence

To achieve a stable convergence the model needs to be complete and accurate. To improve the convergence, accuracy and speed of the simulation, non-zero collector, base, and emitter resistances should be included in the model. When these resistances are not specified, the junction currents are exponentially dependent on the collector, base, and emitter bias voltages. When the bias voltages have no limiting factors, an overflow of current is likely to occur.

Scaling

Scaling of currents, resistances, or capacitances is controlled by the *area* control option. This dimensionless parameter multiplies all currents and capacitors, and divides all resistors. The default value for *area* is 1.0. The DC model parameters `IBC`, `IBE`, `IS`, `ISE`, `ISC`, `IKF`, `IKR`, and `IRB` for both vertical and lateral `BJT` transistors are scaled according to the following equations:

$$IBC_{eff} = area \cdot IBC \quad (10\text{-}9)$$

$$IBE_{eff} = area \cdot IBE \quad (10\text{-}10)$$

$$IS_{eff} = area \cdot IS \quad (10\text{-}11)$$

$$ISE_{eff} = area \cdot ISE \quad (10\text{-}12)$$

$$ISC_{eff} = area \cdot ISC \quad (10\text{-}13)$$

$$IKF_{eff} = area \cdot IKF \quad (10\text{-}14)$$

$$IKR_{eff} = area \cdot IKR \quad (10\text{-}15)$$

$$IRB_{eff} = area \cdot IRB \quad (10\text{-}16)$$

For vertical and lateral devices, the resistor model parameters `RB`, `RBM`, `RE`, and `RC` are scaled according to the following equations:

$$RB_{eff} = RB / area \quad (10\text{-}17)$$

$$RBM_{eff} = RBM / area \quad (10\text{-}18)$$

$$RE_{eff} = RE / area \quad (10\text{-}19)$$

$$RC_{eff} = RC / area \quad (10\text{-}20)$$

BJT Current Convention

Figure 10-4 shows the direction of current flow through a bipolar NPN device. For a bipolar PNP device, polarities of the terminal voltages, gate junction directions, and the direction of the current sources are reversed.

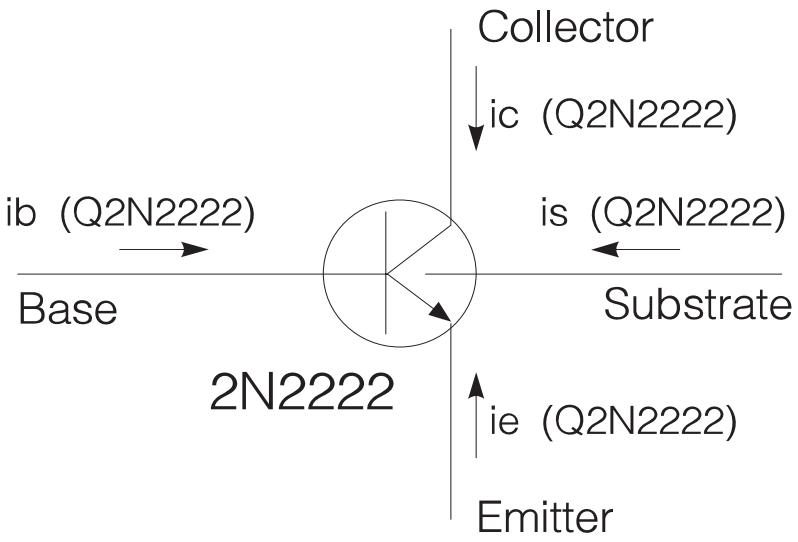

Figure 10-4: NPN BJT Current Convention (PNP are opposite)

BJT Equivalent Circuit

MIXEDMODE uses two equivalent circuits for analysis of bipolar devices: DC and transient. The fundamental components in the equivalent circuit are the collector (i_c) and base (i_b) currents, and their partial derivatives with respect to terminal voltages v_{bc} and v_{be}. The i_c and i_b equations account for all DC effects of the BJT. The names of the partial derivatives are:

Output conductance:

$$g_o = \left.\frac{\partial i_c}{\partial v_{ce}}\right|_{v_{be} = const} = -\left.\frac{\partial i_c}{\partial v_{bc}}\right|_{v_{be} = const} \quad \text{10-21}$$

Transconductance:

$$gm = \left.\frac{\partial i_c}{\partial vbe}\right|_{v_{ce} = const} = \frac{\partial i_c}{\partial v_{be}} + \frac{\partial i_c}{\partial v_{bc}} = \frac{\partial i_c}{\partial vbe} - go \quad \text{10-22}$$

Forward bias conductance:

$$g_\pi = \left.\frac{\partial i_b}{\partial vbe}\right|_{v_{bc} = const} \quad \text{10-23}$$

Reverse bias conductance:

$$g\mu = \left.\frac{\partial ib}{\partial vbc}\right|_{v_{be} = const} \qquad 10\text{-}24$$

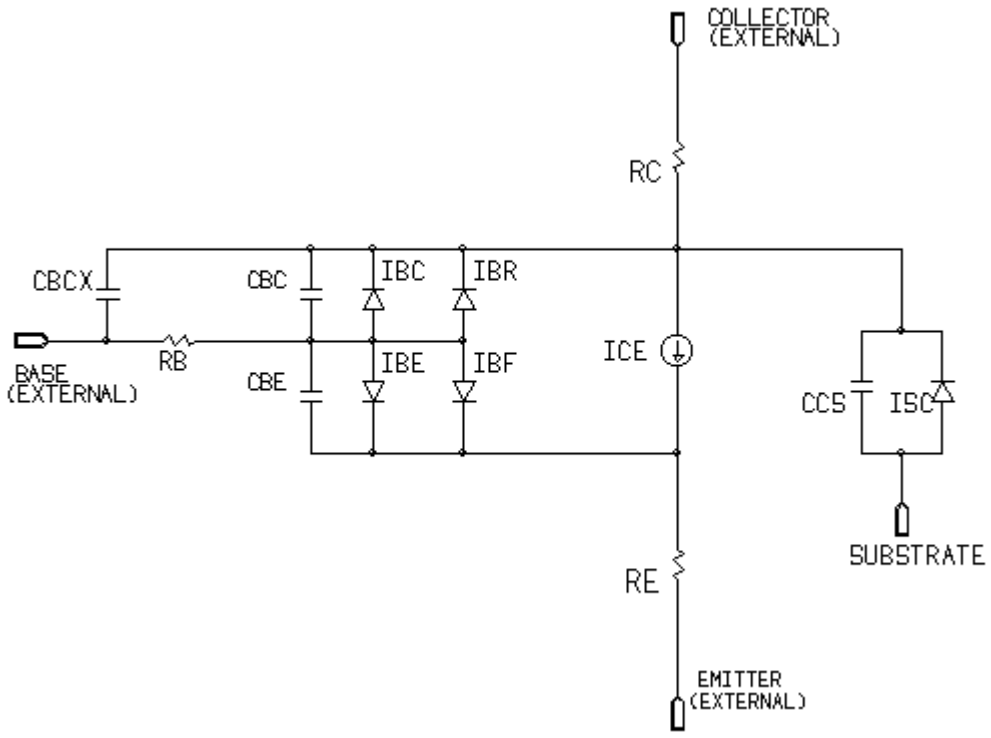

Figure 10-5: BJT Equivalent Circuit for DC and Transient Analysis

Bipolar models are selected by use of both the element and model statements. The element statement references the model statement by the reference model name.

Where the region of operation is narrow, the Ebers-Moll models are sufficiently accurate. The simplicity of the models makes computation efficient. The model parameters are easy to understand and trace back to the process parameters. The simplicity of the Ebers-Moll models comes at a price of many approximations. Wide-application ranges, for example, can create significant errors. The Gummel-Poon model extends over the boundaries of the Ebers-Moll model. Many incorporated second order effects make the model more accurate over all four operating regions: active, reverse, cutoff, and saturated. The Gummel-Poon model is more difficult to understand because it is derived from a physical model. If second order parameters are not specified by the user, the model defaults to a simpler Ebers-Moll model.

The first contribution of the Gummel-Poon model is the inclusion of the finite base, collector, and emitter resistances. These resistances represent the ohmic resistance between the bonding wire or contact and the semiconductor material itself. These resistances change the bias across the internal junctions and have a profound impact on the values of the collector and base currents. The collector resistance affects the performance of transistors at higher frequencies and also limits their current handling capability. The base resistance affects the switching speed. The emitter resistance normally affects the small signal current gain.

An additional contribution of the Gummel-Poon model is the inclusion of accurate models for junction capacitances. These capacitances are represented in the model by corresponding changes in the

emitter, base, and collector junctions. Their junction voltage dependence is modeled by three parameters: CJX, MJX, and VJX.

Base-collector capacitance is modeled with CJC, MJC, and VJC parameters. Base-emitter capacitance is modeled with parameters CJE, MJE, and VJE. Collector-substrate capacitance is modeled with parameters CJS, MJS, and VJS. An additional non-physical parameter, FC, improves the modeling of all junction capacitances under conditions of moderate to large forward bias. It also prevents computing excessive values for capacitances under large forward bias.

For very small values of VBE, secondary order effects such as surface leakage currents and finite resistances make the collector and base currents vary non-linearly with VBE. For large values of VBE, high current injection effects make the collector and base currents non-linear functions of VBE. These effects are modeled by the inclusion of two diodes. In the forward active region, these effects are represented by five parameters: IS, NF, ISE, NE, and IKF. They can be easily extracted from the forward Gummel Plot. The forward Gummel Plot is defined as the plot of collector and base currents versus the intrinsic base emitter voltage. In the reverse region, the parameters BR, NR, ISC, and NC model the reverse Gummel Plot. VAF, the forward early voltage and VAR, the reverse early voltage, are used to model the variations of finite output conductances.

TF is defined as the minimum value of the transit time at high collector-emitter bias. The parameter XTF models the minimum value of TF at high currents and low collector-emitter bias. The parameter ITF models the increase of TF with increasing collector current. The parameter VTF controls the effect of the collector-emitter bias at high collector currents.

The last area where the Gummel-Poon model has made a contribution is the modeling of non-linear effects of the base resistance variation with the base current. The distributed nature of the base resistance makes it very difficult to model it. Three parameters used to model base resistance are RB, IRB, and RBM.

MIXEDMODE models bipolar NPN and PNP devices the same way. In the case of a PNP device, the polarities of the node voltages VBE and VBC are reversed. After computing the collector and base currents, their signs are also reversed for proper polarity.

The DC model is defined by the parameters IS, BF, NF, IKF, ISE, and NE which determine the forward current gain characteristics; IS, BR, NR, IKR, ISC, and NC which determine the reverse current gain characteristics; and VAF and VAR which determine the output conductance for forward and reverse regions. Five ohmic resistances RC, RB, IRB, RBM, and RE are included. Base charge storage is modeled by forward and reverse transit times TF and TR, and nonlinear depletion layer capacitances which are determined by CJE, VJE, and MJE for the B-E junction; CJC, VJC, and MJC for the B-C junction; and CJS, VJS, and MJS for the C-S junction.

The temperature dependence of the saturation current, IS, is determined by the energy gap EG and the saturation current temperature exponent XTI. Base current temperature dependence is modeled by the BF temperature exponent XTB.

A complete Gummel-Poon parameter list, grouped according to basic effects, is:

Table 10-3. A complete Gummel-Poon parameter list

Basic	DC, IBC, IBE, IS, BF, NF, BR, NR
Degradation	ISE, NE, ISC, NC, IKF, IKR
Resistor	RC, RB, RBM, IRB, RE

Table 10-3. A complete Gummel-Poon parameter list	
Junction Capacitor	CJE, VJE, MJE, FC, CJC, VJC, MJC, CJS, VJS, MJS, XCJC
Transition Time	TF, XTF, ITF, VTF,PTF, TR
Temperature Effects	XTB, EG, XTB

BJT Model Parameters

Table 10-4. Basic DC Model Parameters				
Parameter	**Description**	**Units**	**Default**	**Area**
GEOM (SUBS)	Substrate connection selector: Substrate connection selector: vertical geometry lateral geometry Default=1 for NPN Default=2 for PNP			
IBC	Reverse saturation current between A0 base and collector. If IBC and IBE are both specified they are used instead of IS	A		
IBE	Reverse saturation current between A0 base and emitter. If IBC and IBE are both specified they are used instead of IS	A		
IS	Transport saturation current	A	$1.0*10^{-16}$	*
BF	Ideal maximum forward beta		100	
NF	Forward current emission coefficient		1.0	
BR	Ideal maximum reverse beta		1.0	
NR	Reverse current emission coefficient		1.0	

Table 10-5. Low Current Beta Degradation Parameters				
Parameter	**Description**	**Units**	**Default**	**Area**
ISE	B-E leakage saturation current	A	0	*
NE	B-E leakage emission coefficient		1.5	
ISC	B-C leakage saturation current	A	0	*
NC	B-C leakage emission coefficient		1.5	

Table 10-6. Base Width Modulation Parameters				
Parameter	**Description**	**Units**	**Default**	**Area**
VA	F Forward early voltage	V	infinite	
VAR	Reverse early voltage	V	infinite	

Table 10-7. High Current Beta Degradation Effect Parameters				
Parameters	**Description**	**Units**	**Default**	**Area**
IKF	Corner for forward beta high current roll-off		A	infinite*
IKR	Corner for reverse beta high current roll-off		A	infinite*

Table 10-8. Parasitic Resistor Parameters				
Parameters	**Description**	**Units**	**Default**	**Area**
RC	Collector resistance	ohm	0	*
RB	Zero bias base resistance	ohm	0	*
RBM	Minimum base resistance at high currents	ohm	RB	*
IRB	Current where base resistance falls halfway to its min value	A	infinite	*
RE	Emitter resistance	Ohm	0	*

Table 10-9. Junction Capacitor Parameters				
Parameter	**Description**	**Units**	**Default**	**Area**
CAPMOD	Capacitance model selector		1	
CJE	B-E zero-bias depletion capacitance	F	0	*
VJE	B-E built-in potential	V	0.75	
MJE	B-E junction exponential factor		0.33	
FC	Coefficient for forward-bias depletion capacitance formula		0.5	
CJC	B-C zero-bias depletion capacitance	F	0	*
VJC	B-C built-in potential	V	0.75	
MJC	B-C junction exponential factor		0.33	
CJS	Zero-bias collector-substrate-capacitance	F	0	*
VJS	Substrate junction built-in potential	V	0.75	
MJS	Substrate junction exponential factor		0.5	
XCJC	Fraction of B-C depletion1 capacitance connected to internal base node		1	

Table 10-10. Transit Time Parameters				
Parameters	**Description**	**Units**	**Default**	**Area**
TF	Ideal forward transit time	sec	0	
XTF	Coefficient for bias dependence of TF voltage describing VBC		0	
ITF	High-current for effect on TF	A	0	*
VTF	Voltage describing TF dependence of VBC	V	infinite	
PTF	Excess phase at f=1.0/(2·π·TF)	deg	0	
TR	Ideal reverse transit time	sec	0	

Table 10-11. Temperature Effect Parameters				
Parameters	**Description**	**Units**	**Default**	**Area**
XTB	Forward and reverse beta temperature exponent		0	
EG	Energy gap for temperature effect on IS:eV 1.17: Silicon 0.69: Schottky Barrier Diode 0.67: Germanium 1.52: Gallium Arsenide		1.11	
XTI	Temperature exponent for effect on IS3			
GAP1	First bandgap correction factor From Sze, alpha term. 7.02*10-4 - Silicon (old value) 4.73*10-4 - Silicon 4.56*10-4 - Germanium 5.41*10-4 - GaAs	ev/deg	7.02*10-4	
GAP2	First bandgap correction factor From Sze, alpha term 1108-Silicon (old value) 636-Silicon 210-Germanium 204-GaAs	deg		
MJC	Base-collector junction exponent (grading factor)		0.33	
MJE	Base-emitter junction exponent (grading factor)		0.33	
MJS	Substrate junction exponent (grading factor)		0.5	
TBF1	First order temp. coefficient for BF	1/deg	0	
TBF2	Second order temp. coefficient for BF	$1/deg^2$	0	
TBR1	First order temp. coefficient for BR	1/deg	0	
TBR2	Second order temp. coefficient for BR	$1/deg^2$	0	
TIKF1	First order temp. coefficient for IKF	1/deg	0	

Table 10-11. Temperature Effect Parameters

Parameters	Description	Units	Default	Area
TIKF2	Second order temp. coeffi-cient for IKF	$1/deg^2$	0	
TIKR1	First order temp. coefficient for IKR	1/deg	0	
TIKR2	Second order temp. coeffi-cient for IKRr	$1/deg^2$	0	
TIRB1	First order temp. coefficient for IRB	1/deg		
TIRB2	Second order temp. coeffi-cient for IRB	$1/deg^2$	0	
TISC1	First order temp. coefficient for ISC. TEMPLEV=3 enables TISC1.	1/deg	0	
TISC2	Second order temp. coeffi-cient for ISC. TEMPLEV=3 enables ISC2.	$1/deg^2$	0	
TIS1	First order temp. coefficient for IS or IBE and IBC. TEM-PLEV=3 enables TIS1.	1/deg	0	
TIS2	Second order temp. coeffi-cient For IS or IBE or IBC. TEMPLEV=3 enables IS2.	$1/deg^2$	0	
TISE1	First order temp. coefficient for ISE. TEMPLEV=3 enables TISE1.	1/deg	0	
TISE2	Second order temp. coeffi-cient for SE. TEMPLEV=3 enables ISE2.	$1/deg^2$	0	
TISS1	First order temp. coefficient for ISS. TEMPLEV=3 enables TISS1.	1/deg	0	
TISS2	Second order temp. coeffi-cient for ISS. TEMPLEV=3 enables ISS2.	$1/deg^2$	0	
TITF1	First order temp. coefficient for ITF	1/deg	0	
TITF2	Second order temp. coeffi-cient for ITF	$1/deg^2$	0	
TEMPLEV	Temperature equation level selector		0	

Table 10-11. Temperature Effect Parameters				
Parameters	**Description**	**Units**	**Default**	**Area**
TEMPLEVC	Temperature equation level selector for junction capacitances and contact potentials		0	
TMJC1	First order temp. coefficient for MJC	1/deg		
TMJC2	Second order temp. coefficient for MJC	$1/deg^2$	0	
TMJE1	First order temp. coefficient for MJE	1/deg	0	
TMJE2	Second order temp. coefficient for MJE	$1/deg^2$	0	
TMJS1	First order temp. coefficient for MJS	1/deg	0	
TMJS2	Second order temp. coefficient for MJS	$1/deg^2$	0	
TNC1	First order temp. coefficient for NC	1/deg	0	
TNC2	Second order temp. coefficient for NC	$1/deg^2$	0	
TNE1	First order temp. coefficient for NE	1/deg	0	
TNE2	Second order temp. coefficient for NE	$1/deg^2$	0	
TNF1	First order temp. coefficient for NF	1/deg	0	
TNF2	Second order temp. coefficient for NF	$1/deg^2$	0	
TNR1	First order temp. coefficient for NR	1/deg	0	
TNR2	Second order temp. coefficient for NR	$1/deg^2$	0	
TNS1	First order temp. coefficient for NS	1/deg	0	
TNS2	Second order temp. coefficient for NS	$1/deg^2$	0	
TRB1	First order temp. coefficient for RB	1/deg	0	
TRB2	Second order temp. coefficient for RB	$1/deg^2$	0	

Table 10-11. Temperature Effect Parameters				
Parameters	**Description**	**Units**	**Default**	**Area**
TRC1	First order temp. coefficient for RC	1/deg	0	
TRC2	Second order temp. coefficient for RC	$1/deg^2$	0	
TRE1	First order temp. coefficient for RE	1/deg	0	
TRE2	Second order temp. coefficient for RE	$1/deg^2$	0	
TRM1	First order temp. coefficient for RM	1/deg	0	
TRM2	Second order temp. coefficient for RM	$1/deg^2$	0	
TTF1	First order temp. coefficient for TF	1/deg	0	
TTF2	Second order temp. coefficient for TF	$1/deg^2$	0	
TTR1	First order temp. coefficient for TTR	1/deg	0	
TTR	Second order temp. coefficient for TTR	$1/deg^2$	0	
TVAF1	First order temp. coefficient for VAF	1/deg	0	
TVAF2	Second order temp. coefficient for VAF	$1/deg^2$	0	
TVAR1	First order temp. coefficient for VAR	1/deg	0	
TVAR2	Second order temp. coefficient for VAR	$1/deg^2$	0	
TVJC	Temp. coefficient for	V/deg	0	
TVJE	Temp. coefficient for	V/deg	0	
TVJS	Temp. coefficient for	V/deg	0	
tnom	Parameter measurement ^{O}C temperature 27			

Lateral and vertical transistors are shown in Figures 10-6 and 10-7.

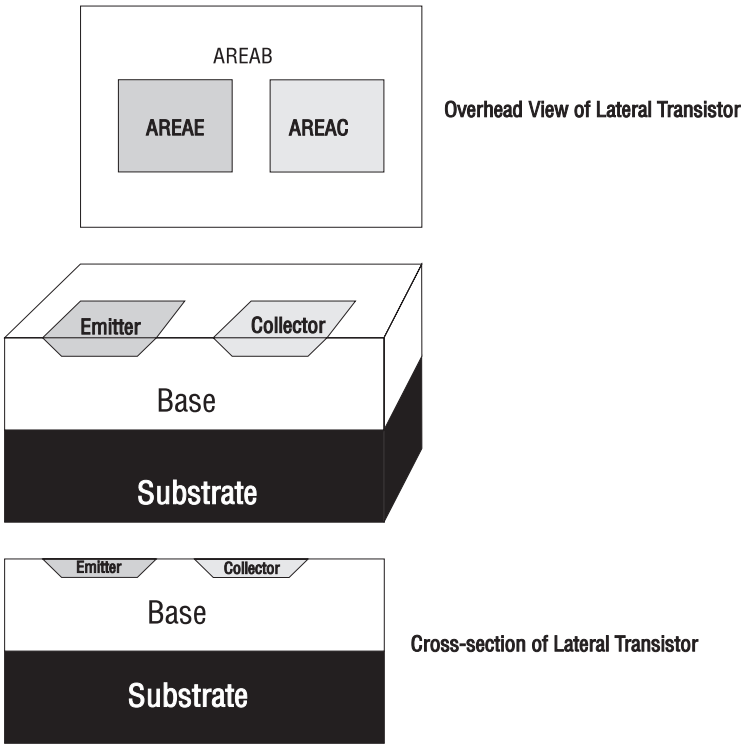

Figure 10-6: A Lateral Transistor

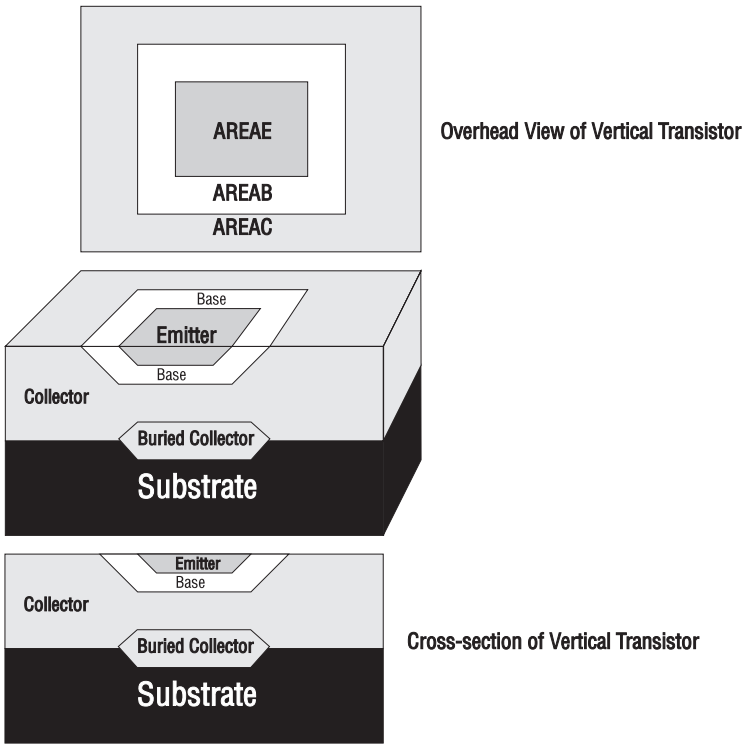

Figure 10-7: A Vertical Transistor

BJT Model Equations

The DC characteristics of the BJT transistor are determined by twenty model parameters.Define:

$$i_{bf} = \text{forward diffusion current} = \frac{IS_{eff}}{BF} \cdot \left(e^{\frac{v_{be}}{\overline{NF \cdot v_t}}} - 1 \right) \quad \text{10-25}$$

$$i_{be} = \text{non-ideal base-emitter current} = ISE \cdot \left(e^{\frac{v_{be}}{\overline{NE \cdot v_t}}} - 1 \right) \quad \text{10-26}$$

$$i_{br} = \text{reverse diffusion current} = \frac{IS_{eff}}{BR} \left(e^{\frac{v_{bc}}{\overline{NR \cdot v_t}}} - 1 \right) \quad \text{10-27}$$

$$i_{bc} = \text{non-ideal base-collector current} = ISC \cdot \left(e^{\frac{v_{bc}}{\overline{NC \cdot v_t}}} - 1 \right) \quad \text{10-28}$$

DC Current Equations

Current equations for the collector current (i_c) and the base current (i_b) are:

$$i_b = i_{bf} + i_{be} + i_{br} + i_{bc} \quad \text{10-29}$$

$$i_c = \frac{IS_{eff}}{qb} \cdot \left(e^{\frac{v_{be}}{NF \cdot v_t}} - e^{\frac{v_{bc}}{NR \cdot v_t}} \right) - \frac{IS_{eff}}{BR} \cdot \left(e^{\frac{v_{bc}}{NR \cdot v_t}} - 1 \right) \quad \text{10-30}$$

$$-ISC_{eff} \cdot \left(e^{\frac{v_{bc}}{NC \cdot v_t}} - 1 \right)$$

If IBE and IBC are both specified instead of IS:

$$i_c = \frac{IBE_{eff}}{qb} \cdot \left(e^{\frac{v_{be}}{NF \cdot v_t}} - 1 \right) - \frac{IBC_{eff}}{qb} \cdot \left(e^{\frac{v_{bc}}{NR \cdot v_t}} - 1 \right) \quad \text{10-31}$$

$$-\frac{IBC_{eff}}{BR} \cdot \left(e^{\frac{v_{bc}}{NR \cdot v_t}} - 1 \right) - ISC_{eff} \cdot \left(e^{\frac{v_{bc}}{NC \cdot v_t}} - 1 \right)$$

$$i_b = \frac{IBE_{eff}}{BF} \cdot \left(e^{\frac{v_{be}}{NF \cdot v_t}} - 1 \right) - \frac{IBC_{eff}}{BR} \cdot \left(e^{\frac{v_{bc}}{NR \cdot v_t}} - 1 \right) \quad \text{10-32}$$

$$+ ISE_{eff} \cdot \left(e^{\frac{v_{bc}}{NR \cdot v_t}} - 1 \right) - ISC_{eff} \cdot \left(e^{\frac{v_{bc}}{NC \cdot v_t}} - 1 \right)$$

Base Charge Equations

The parameters VAF and VAR are forward and reverse early voltage. The parameters IKF and IKR determine the high current BF roll-off. ISE, NI, ISC, and NC determine the low current BF roll-off.

qb = base charge factor 10-33

$$q_1 = \frac{1}{\left(1 - \frac{v_{bv}}{VAF} - \frac{v_{be}}{VAR} \right)} \quad \text{10-34}$$

$$q_2 = \frac{ISE_{eff}}{IKF_{eff}} \cdot \left(e^{\frac{v_{be}}{NF \cdot v_t}} - 1 \right) + \frac{ISC_{eff}}{IKR_{eff}} \cdot \left(e^{\frac{v_{bc}}{NR \cdot v_t}} - 1 \right) \quad \text{10-35}$$

$$q_b = \frac{q_1}{2} \cdot \left[1 + (1 + 4 \cdot q_2)^{1/2} \right] \quad \text{10-36}$$

Substrate Current Equations

The substrate current is substrate to collector for vertical transistors and substrate to base for lateral transistors.

is = substrate current = $$area \cdot ISS \cdot (e^{v_{jsover}(NS \cdot v_t)} - 1)$$ 10-37

For vertical transistors

isc = $$ISS_{eff} \cdot \left(e^{\frac{v_{sc}}{NS \cdot v_t}} - 1 \right)$$ when v sub sc > -10 · NS · vt 10-38

isc = ISS_{eff} when vsc ≤ –10 · ns · VT 10-39

For lateral transistors:

i_{bs} = $$ISS_{eff} \cdot \left(e^{\frac{vbs}{NS \cdot v_t}} - 1 \right)$$ when vbs > -10 · NS · v sub t 10-40

ibs = ISSeff when vbs ≤ -10 · NS · vt 10-41

If both IBE and IBC are unspecified:

ISS_{eff} = ISS · area 10-42

Variable Base Resistance Equations

MIXEDMODE provides a variable base resistance model that consists of a low-current maximum resistance set by RB and a high-current minimum resistance set by RBM. IRB is the current at which the base resistance falls halfway to its minimum value. If RBM is not specified, it is defaulted to RB.

If IRB is not specified:

$$rbb = RBM_{eff}\left(\frac{RB_{eff} + RBM_{eff}}{qb}\right)$$ 10-43

If IRB is specified:

$$rbb = RBM_{eff} + 3 \cdot (RB_{eff} - RBM_{eff}) \cdot \frac{\tan(z) - z}{z \cdot \tan(z) \cdot \tan(z)}$$ 10-44

where:

$$z = \frac{-1 + \left[1 + 144 \cdot \frac{i_b}{(\pi^2 \cdot IRB_{eff})}\right]^{1/2}}{\frac{24}{\pi^2} \cdot \left(\frac{i_b}{IRB_{eff}}\right)^{1/2}}$$ 10-45

Capacitance Equations

The total capacitance for all junctions contains two terms: depletion capacitance and diffusion capacitance. The depletion capacitance dominates low frequency behavior. The diffusion capacitance dominates high frequency behavior.

Base Emitter Capacitance Equations

The base-emitter capacitance contains a complex diffusion term with a standard depletion capacitance formula. The diffusion capacitance is modified by model parameters TF, XTF, ITF, and VTF.

The base emitter capacitance is determined by:

$$\texttt{cbe} = \texttt{cbediff} + \texttt{cbedep} \qquad \text{10-46}$$

where cbediff is the base-emitter diffusion capacitance and cbedep is the base-emitter depletion capacitance.

Base-Emitter Diffusion Capacitance

$$cbediff = \frac{\partial}{\partial v_{BE}}\left(TF \cdot \frac{ibe}{qb}\right) \text{ for ibe} \le 0 \qquad \text{10-47}$$

$$cbediff = \frac{\partial}{\partial v_{be}}\left[TF \cdot (1 + \arg tf) \cdot \frac{ibe}{qb}\right] \text{ for ibe} > 0 \qquad \text{10-48}$$

where:

$$\texttt{argtf} = XTF \cdot \left(\frac{ibe}{ibe + ITF}^{2}\right) \cdot e^{\frac{v_{bc}}{144 \cdot VTF}} \qquad \text{10-49}$$

Internal base-emitter current is determined by:

$$\text{ibe} = ISE_{eff} \cdot \left(\frac{v_{be}}{e^{\frac{NE \cdot v_t}{}} - 1}\right) \qquad \text{10-50}$$

Base-Emitter Depletion Capacitance

There are two different equations for modeling the depletion capacitance. The desired equation is selected by specifying the option CAPMOD in the .OPTIONS statement.

CAPMOD=1

$$cbedep = CJE_{eff} \cdot \left(1 - \frac{v_{be}}{VJE}\right)^{-MJE} \quad \text{for vbe} < \text{FC} \cdot \text{VJE} \qquad \text{10-51}$$

$$cbedep = CJE_{eff} \cdot \frac{1 - FC \cdot (1 + MJE) + MJE \cdot \frac{v_{be}}{VJE}}{(1 - FC)^{(1 + MJE)}} \quad \text{for } v_{be} \ge \text{FC} \cdot \text{VJE} \qquad \text{10-52}$$

CAPMOD=2

The base-emitter depletion capacitance is determined as follows:

$$cbedep = CJE_{eff} \cdot \left(1 - \frac{v_{be}}{VJE}\right)^{-MJE} \quad \text{for } v_{be} < 0 \qquad 10\text{-}53$$

$$cbedep = CJE_{eff} \cdot \left(1 + MJE \cdot \frac{v_{be}}{VJE}\right) \text{ for } v_{be} \geq 0 \qquad 10\text{-}54$$

Base-Collector Capacitance Equations

The base-collector capacitance contains a complex diffusion term with a standard depletion formula. The diffusion capacitance is modified by model parameter, TR.

The base-collector capacitance cbc is determined by:

```
cbc = cbcdiff + cbcdep
```  10-55

where cbcdiff is the base-collector diffusion capacitance and cbcdep is the depletion capacitance.

#### *Base-Collector Diffusion Capacitance*

$$cbcdiff = \frac{\partial}{\partial_{v_{bc}}}(TR \cdot ibc) \qquad 10\text{-}56$$

where the internal base-collector current ibc is:

$$ibc = IS_{eff} \cdot \left(e^{\frac{v_{bc}}{NR \cdot v_t}} - 1\right) \qquad 10\text{-}57$$

#### *Base-Collector Depletion Capacitance*

There are two different equations for modeling the depletion capacitance. Select the desired equations by specifying the option, CAPMOD in the .OPTIONS statement.

CAPMOD=1

$$cbcdep = XCJC \cdot CJC_{eff} \cdot \left(1 - \frac{v_{bc}}{VJC}\right)^{-MJC} \quad \text{for } v_{bc} < FC \cdot VJC \qquad 10\text{-}58$$

$$cbcdep = XCJC \cdot CJC_{eff} \cdot \frac{1 - FC \cdot (1 + MJC) + MJC \cdot \frac{v_{bc}}{VJC}}{(1 - FC)^{(1 + MJC)}} \qquad 10\text{-}59$$

**CAPMOD=2**

$$cbcdep = XCJC \cdot CJC_{eff} \cdot \left(1 - \frac{v_{bc}}{VJC}\right) \text{ for } v_{bc} < 0 \qquad 10\text{-}60$$

$$cbcdep = XCJX \cdot CJC_{eff} \cdot \left(1 + MJC \cdot \frac{v_{bc}}{VJC}\right) \text{ for } v_{bc} \geq 0 \qquad 10\text{-}61$$

#### *External Base-Internal Collector Junction Capacitance*

The base-collector capacitance can be modeled as a distributed capacitance by setting parameter, XCJC. When XCJC is one (set or defaulted), the entire base-collector capacitance is connected to the internal base node.

**CAPMOD=1** Assuming FC = 0:

$$cbcx = CJC_{eff} \cdot (1 - XCJC) \cdot \left(1 - \frac{vbcx}{VJC}\right)^{-MJC} \quad \text{for vbcx} < FC \cdot VJC \qquad 10\text{-}62$$

$$cbcx = CJC_{eff} \cdot (1 - XCJC) \cdot \frac{1 - FC \cdot (1 + MJC) + MJC \cdot \frac{vbcx}{VJC}}{(1 - FC)^{(1 + MJC)}} \qquad 10\text{-}63$$

for vbcx $\geq$ FC · VJC

**CAPMOD=2**

$$cbcx = CJC_{eff} \cdot (1 - XCJC) \cdot \left(1 - \frac{vbcx}{VJC}\right)^{-MJC} \quad \text{for vbcx} < FC \cdot VJC \qquad 10\text{-}64$$

$$cbcx = CJC_{eff} \cdot (1 - XCJC) \cdot \left(1 + MJC \cdot \frac{vbcx}{VJC}\right) \quad \text{for vbcx} \geq 0 \qquad 10\text{-}65$$

where vbcx is the voltage between the external base node and the internal collector node.

#### *Substrate Capacitance*

#### *For Lateral Substrate Capacitance:*

For reverse bias:

$$cbs = CJS_{eff} \cdot \left(1 - \frac{vbs}{VJS}\right)^{-MJS} \quad \text{for } v_{bs} < 0 \qquad 10\text{-}66$$

For forward bias:

$$cbs = CJS_{eff} \cdot \left(1 + MJS \cdot \frac{vbs}{VJS}\right) \quad \text{for vbs} \geq 0 \qquad 10\text{-}67$$

#### *For Vertical Substrate Capacitance:*

For reverse bias:

$$\text{csc} = CJS_{eff} \cdot \left(1 - \frac{v_{sc}}{VJS}\right)^{-MJS} \quad \text{for } v_{sc} < 0 \qquad 10\text{-}68$$

For forward bias:

$$\text{csc} = CJS_{eff} \cdot \left(1 + MJS \cdot \frac{v_{sc}}{VJS}\right) \qquad 10\text{-}69$$

## Temperature Effects Equations

### *Analysis at Different Temperatures*

All model input data for MIXEDMODE are assumed to have been extracted at a nominal temperature of 27°C. The circuit simulation will also be performed at a temperature of 27°C, unless overridden by a TEMP parameter on the .MODEL control statement.

TEMPLEV=1

The following equations model the energy gap with temperature:

$$egnom = 1.16 - 7.02e-4 \cdot \frac{tnom^2}{tnom + 1108.0} \quad \text{10-70}$$

$$eg(t) = 1.16 - 7.02e-4 \cdot \frac{t^2}{t + 1108.0} \quad \text{10-71}$$

The basic BJT temperature compensation equations for betas and the saturation currents are:

$$BF(t) = BF \cdot \left(\frac{t}{tnom}\right)^{XTB} \quad \text{10-72}$$

$$BR(t) = BR \cdot \left(\frac{t}{tnom}\right)^{XTB} \quad \text{10-73}$$

$$ISE(t) = \frac{ISE}{\left(\frac{t}{tnom}\right)^{XTB}} \cdot e^{\frac{facln}{NE}} \quad \text{10-74}$$

$$ISC(t) = \frac{ISC}{\left(\frac{t}{tnom}\right)^{XTB}} \cdot e^{\frac{facln}{NC}} \quad \text{10-75}$$

$$ISS(t) = \frac{ISS}{\left(\frac{t}{tnom}\right)^{XTB}} \cdot e^{\frac{facln}{NS}} \quad \text{10-76}$$

$$IS(t) = IS \cdot e^{fac1n} \quad \text{10-77}$$

$$IBE(t) = IBE \cdot e^{\frac{fac1n}{NF}} \quad \text{10-78}$$

$$IBC(t) = IBC \cdot e^{\frac{fac1n}{NR}} \quad 10\text{-}79$$

where:

$$fac1n = \frac{EG}{v_{t(tnom)}} - \frac{EG}{v_{t(t)}} + XTI \cdot \ln\left(\frac{t}{tnom}\right) \quad 10\text{-}80$$

$$IKF(t) = IKF \cdot (1 + TIKF1 \cdot \Delta t + TIKF2 \cdot \Delta t^2) \quad 10\text{-}81$$

$$IKR(t) = IKR \cdot (1 + TIKR1 \cdot \Delta t + TIKR2 \cdot \Delta t^2) \quad 10\text{-}82$$

$$IRB(t) = IRB \cdot (1 + TIRB1 \cdot \Delta t + TIRB2 \cdot \Delta t^2) \quad 10\text{-}83$$

TEMPLEV=2

The following equations model the energy gap with temperature:

$$egnom = 1.16 - 7.02e-4 \cdot \frac{tnom^2}{tnom + 1108.0} \quad 10\text{-}84$$

$$eg(t) = 1.16 - 7.02e-4 \cdot \frac{t^2}{t + 1108.0} \quad 10\text{-}85$$

The basic BJT temperature compensation equations for betas and the saturation currents are:

$$BF(t) = BF \cdot (1 + XTB \cdot \Delta t) \quad 10\text{-}86$$

$$BR(t) = BR \cdot (1 + XTB \cdot \Delta t) \quad 10\text{-}87$$

$$ISE(t) = \frac{ISE}{1 + XTB \cdot \Delta t} \cdot e^{\frac{fac1n}{NE}} \quad 10\text{-}88$$

$$ISC(t) = \frac{ISC}{1 + XTB \cdot \Delta t} \cdot e^{\frac{fac1n}{NC}} \quad 10\text{-}89$$

$$ISS(t) = \frac{ISA}{1 + XTB \cdot \Delta t} \cdot e^{\frac{fac1n}{NE}} \quad 10\text{-}90$$

$$IS(t) = IS \cdot e^{fac1n} \quad 10\text{-}91$$

$$IBE(t) = IBE \cdot e^{\frac{fac1n}{NF}} \quad 10\text{-}92$$

$$IBC(t) = IBC \cdot e^{\frac{fac1n}{NR}} \quad 10\text{-}93$$

where:

$$fac1n = \frac{EG}{v_{t(tnom)}} - \frac{EG}{v_{t(t)}} + XTI \cdot \ln\left(\frac{t}{tnom}\right) \quad 10\text{-}94$$

$$IKF(t) = IKF \cdot (1 + TIKF1 \cdot \Delta t + TIKF2 \cdot \Delta t^2) \quad 10\text{-}95$$

$$IKR(t) = IKR \cdot (1 + TIKR1 \cdot \Delta t + TIKR2 \cdot \Delta t^2) \quad 10\text{-}96$$

$$IRB(t) = IRB \cdot (1 + TIRB1 \cdot \Delta t + TIRB2 \cdot \Delta t^2) \quad 10\text{-}97$$

TEMPLEV=3

The following equations model the energy gap with temperature:

$$egnom = EG - GAP1 \cdot \frac{tnom^2}{tnom + GAP2} \quad 10\text{-}98$$

$$eg(t) = EG - GAP1 \cdot \frac{t^2}{t + GAP2} \quad 10\text{-}99$$

The basic BJT temperature compensation equations for betas and the saturation currents are:

$$BF(t) = BF \cdot \left(\frac{t}{tnom}\right)^{XTB} \quad 10\text{-}100$$

$$BR(t) = BR \cdot \left(\frac{t}{tnom}\right)^{XTB} \quad 10\text{-}101$$

$$ISE(t) = \frac{ISE}{\left(\frac{t}{tnom}\right)^{XTB}} \cdot e^{\frac{fac\ln}{NE}} \quad 10\text{-}102$$

$$ISC(t) = \frac{ISC}{\left(\frac{t}{tnom}\right)^{XTB}} \cdot e^{\frac{fac\ln}{NC}} \quad 10\text{-}103$$

$$ISS(t) = \frac{ISS}{\left(\frac{t}{tnom}\right)^{XTB}} \cdot e^{\frac{fac\ln}{NS}} \quad 10\text{-}104$$

$$IS(t) = IS \cdot e^{fac1n} \quad 10\text{-}105$$

$$IBE(t) = IBE \cdot e^{\frac{fac1n}{NF}} \quad 10\text{-}106$$

$$IBC(t) = IBC \cdot e^{\frac{fac1n}{NR}} \quad \text{10-107}$$

where:

$$fac1n = \frac{EG}{v_{t(tnom)}} - \frac{EG}{v_{t(t)}} + XTI \cdot \ln\left(\frac{t}{tnom}\right) \quad \text{10-108}$$

$IKF(t) = IKF \cdot (1 + TIKF1 \cdot \Delta t + TIKF2 \cdot \Delta t2)$ 10-109

$IKF(t) = IKF \cdot (1 + TIKF1 \cdot \Delta t + TIKF2 \cdot \Delta t2)$ 10-110

$IRB(t) = IRB \cdot (1 + TIRB1 \cdot \Delta t + TIRB2 \cdot \Delta t2)$ 10-111

TEMPLEV=4

The basic BJT temperature compensation equations for the saturation currents when TEMPLEV=4 are:

$IS(t) = IS\ (1+TIS1 \cdot \Delta t + TIS2 \cdot \Delta t2).$ 10-112

$IBE(t) = IBE(1+TIS1 \cdot \Delta t + TIS2 \cdot \Delta t2)$ 10-113

$IBC(t) = IBC(1+TIS1 \cdot \Delta t + TIS2 \cdot \Delta t2).$ 10-114

$ISE(t) = ISE(1+TISE1 \cdot \Delta t + TISE2 \cdot \Delta t2)$ 10-115

$ISE(t) = ISE(1+TISE1 \cdot \Delta t + TISE2 \cdot \Delta t2)$ 10-116

$ISC(t) = ISC(1+TISC1 \cdot \Delta t + TISC2 \cdot \Delta t2)$ 10-117

$ISS(t) = ISS(1+TISS1 \cdot \Delta t + TISS2 \cdot \Delta t2)$ 10-118

The parameters IKF, IKR, and IRB are modified as follows:

$IKF(t) = IKF(1+TIKF1 \cdot \Delta t + TIKF2 \cdot \Delta t2)$ 10-119

$IKR(t) = IKR(1+TIKR1 \cdot \Delta t + TIKR2 \cdot \Delta t2)$ 10-120

$IRB(t) = IRB(1+TIRB1 \cdot \Delta t + TIRB2 \cdot \Delta t2)$ 10-121

The following parameters are modified when corresponding temperature coefficients are specified, regardless of the value of the parameter TEMPLEV:

$BF(t) = BF \cdot (1 + TBF1 \cdot \Delta t + TBF2 \cdot \Delta t2)$ 10-122

$BR(t) = BR \cdot (1 + TBR1 \cdot \Delta t + TBR2 \cdot Dt2)$ 10-123

$VAF(t) = VAF \cdot (1 + TVAF1 \cdot \Delta t + TVAF2 \cdot \Delta t2)$ 10-124

$VAR(t) = VAR \cdot (1 + TVAR1 \cdot \Delta t + TVAR2 \cdot \Delta t2)$ 10-125

$ITF(t) = ITF \cdot (1 + TITF1 \cdot \Delta t + TITF2 \cdot \Delta t2)$ 10-126

$TF(t) = TF \cdot (1 + TTF1 \cdot \Delta t + TTF2 \cdot \Delta t2)$ 10-127

$TR(t) = TR \cdot (1 + TTR1 \cdot Dt + TTR2 \cdot Dt2)$ 10-128

$NF(t) = NF \cdot (1 + TNF1 \cdot \Delta t + TNF2 \cdot \Delta t2)$ 10-129

$NR(t) = NR \cdot (1 + TNR1 \cdot \Delta t + TNR2 \cdot \Delta t2)$ 10-130

$NE(t) = NE \cdot (1 + TNE1 \cdot \Delta t + TNE2 \cdot \Delta t2)$ 10-131

$NC(t) = NC \cdot (1 + TNC1 \cdot \Delta t + TNC2 \cdot \Delta t2)$ 10-132

$NS(t) = NS \cdot (1 + TNS1 \cdot \Delta t + TNS2 \cdot \Delta t2)$ 10-133

$MJE(t) = MJE \cdot (1 + TMJE1 \cdot \Delta t + TMJE2 \cdot \Delta t2)$ 10-134

$MJC(t) = MJC \cdot (1 + TMJC1 \cdot \Delta t + TMJC2 \cdot \Delta t2).$ 10-135

$MJS(t) = MJS \cdot (1 + TMJS1 \cdot \Delta t + TMJS2 \cdot \Delta t2)$ 10-136

### Capacitance Temperature Equations

TEMPLEVC=1

$$CJE(t) = CJE \cdot \left[1 + MJE \cdot \left(0.0004 \cdot \Delta t - \frac{VJE(t)}{VJE} + 1\right)\right] \quad \text{10-137}$$

$$CJC(t) = CJC \cdot \left[1 + MJC \cdot \left(0.0004 \cdot \Delta t - \frac{VJC(t)}{VJC} + 1\right)\right] \quad \text{10-138}$$

$$CJS(t) = CJS \cdot \left[1 + MJS \cdot \left(0.0004 \cdot \Delta t - \frac{VJS(t)}{VJS} + 1\right)\right] \quad \text{10-139}$$

where:

$$VJE(t) = VJE \cdot \frac{t}{tnom} - v_{t(t)} \cdot \left[3 \cdot 1n\left(\frac{t}{tnom}\right) + \frac{egnom}{v_{t(tnom)}} - \frac{eg(t)}{v_{t(t)}}\right] \quad \text{10-140}$$

$$VJC(t) = VJC \cdot \frac{t}{tnom} - v_{t(t)} \cdot \left[3 \cdot 1n\left(\frac{t}{tnom}\right) + \frac{egnom}{v_{t(tnom)}} - \frac{eg(t)}{v_{t(t)}}\right] \quad \text{10-141}$$

$$VJS(t) = VJS \cdot \frac{t}{tnom} - v_{t(t)} \cdot \left[3 \cdot 1n\left(\frac{t}{tnom}\right) + \frac{egnom}{v_{t(tnom)}} - \frac{eg(t)}{v_{t(t)}}\right] \quad \text{10-142}$$

TEMPLEVC=2

```
CJE(t) = CJE · (1 + CTE · Dt)                    10-143
CJC(t) = CJC · (1 + CTC · Dt)                    10-144
CJS(t) = CJS · (1 + CTS · Dt)                    10-145
```

and contact potentials are determined as:

```
VJE(t) = VJE · (1 + TVJE · Dt)                   10-146
VJC(t) = VJC · (1 + TVJC · Dt)                   10-147
VJS(t ) =  VJS ·  (1 + TVJS · Dt)                10-148
```

TEMPLEVC=3

$$CJE(t) = CJE \cdot \left(\frac{VJE}{VJE(t)}\right)^{MJE} \quad \text{10-149}$$

$$CJC(t) = CJC \cdot \left(\frac{VJC}{VJC(t)}\right)^{MJC} \quad \text{10-150}$$

$$CJS(t) = CJS \cdot \left(\frac{VJS}{VJS(t)}\right)^{MJS} \quad \text{10-151}$$

where:

VJE(t) = VJE - TVJE · Dt 10-152

VJC(t) = VJC - TVJC · Dt 10-153

VJS(t) = VJS - TVJS · Dt 10-154

TEMLEVC=4

$$CJE(t) = CJE \cdot \left(1 - 0.5 \cdot dvjedt \cdot \frac{\Delta t}{VJE}\right) \quad 10\text{-}155$$

$$CJC(t) = CJC \cdot \left(1 - 0.5 \cdot dvjcdt \cdot \frac{\Delta t}{VJC}\right) \quad 10\text{-}156$$

$$CJS(t) = CJS \cdot \left(1 - 0.5 \cdot dvjsdt \cdot \frac{\Delta t}{VJS}\right) \quad 10\text{-}157$$

VJE(t) = VJE + dvjedt · Δt 10-158

VJC(t) = VJC + dvjcdt · Δ 10-159

VJS(t) = VJS + dvjsdt · Δ 10-160

where for TEMPLEV=1 or 2:

dvjedt = egnom = $3 \cdot v_{t(tnom)}$ + (1.16 – egnom)

$$\cdot \left(2 - \frac{tnom}{\frac{tnom + 1108}{tnom}}\right) - VJE \quad 10\text{-}161$$

dvjcdt = egnom = $3 \cdot v_{t(tnom)}$ + (1.16 – egnom)

$$\cdot \left(2 - \frac{tnom}{\frac{tnom + 1108}{tnom}}\right) - VJC \quad 10\text{-}162$$

dvjsdt = egnom = $3 \cdot v_{t(tnom)}$ + (1.16 – egnom)

$$\cdot \left(2 - \frac{tnom}{\frac{tnom + 1108}{tnom}}\right) - VJS \quad 10\text{-}163$$

and for TEMPLEV=3:

dvjedt = egnom = $3 \cdot v_{t(tnom)}$ + (EG – egnom)

$$\cdot \left(2 - \frac{tnom}{\frac{tnom + GAP2}{tnom}}\right) - VJE \quad 10\text{-}164$$

$$dvjcdt = egnom = 3 \cdot v_{t(tnom)} + (EG - egnom) \cdot \left(2 - \frac{tnom}{\frac{tnom + GAP2}{tnom}} \right) - VJC \quad 10\text{-}165$$

$$dvjsdt = egnom = 3 \cdot v_{t(tnom)} + (EG - egnom) \cdot \left(2 - \frac{tnom}{\frac{tnom + 1108GAP2}{tnom}} \right) - VJS \quad 10\text{-}166$$

Parasitic Resistor Temperature Equations

The following parameters are also modified when corresponding temperature coefficients are specified:

$$RE(T) = RE \cdot (1 + TRE1 \cdot (t - tnom) + TRE2 \cdot (t - tnom)2 \quad 10\text{-}167$$

$$RB(t) + RB \cdot (1 + TRE1 \cdot (t - tnom) + \quad 10\text{-}168$$

MOSFET Model

MIXEDMODE supports the popular LEVEL 3 model which is widely used in SPICE circuit analysis programs.

MOSFETs

The equations inside MIXEDMODE are the same for both n- and p-channel devices, and MIXEDMODE evaluates the current equations for n- and p-channel devices in the same way. For p-channel devices, the polarities of the threshold voltage and the node voltages VGS, VDS, and VBS are reversed, and the transistor is handled internally as an n-channel device. The final p-channel drain current is then obtained by reversing the sign of the computed drain current.

Scaling

Scaling of currents, resistances, or capacitances is controlled by a number of parameters (see the description of the MXXX statement). These parameters are:

AD	Area of drain
AS	Area of source
NRD	Number of squares for drain resistance
NRS	Number of squares for source resistance
PD	Sidewall of drain
PS	Sidewall of source

MOSFET Current Convention

Current flow through the MOSFET is assumed to be in accordance with the following diagram. Figure 10-9 shows the current convention for an n-channel MOSFET. For a p-channel MOSFET, polarities of the terminal voltages, gate junction directions, and the direction of the nonlinear current source are reversed.

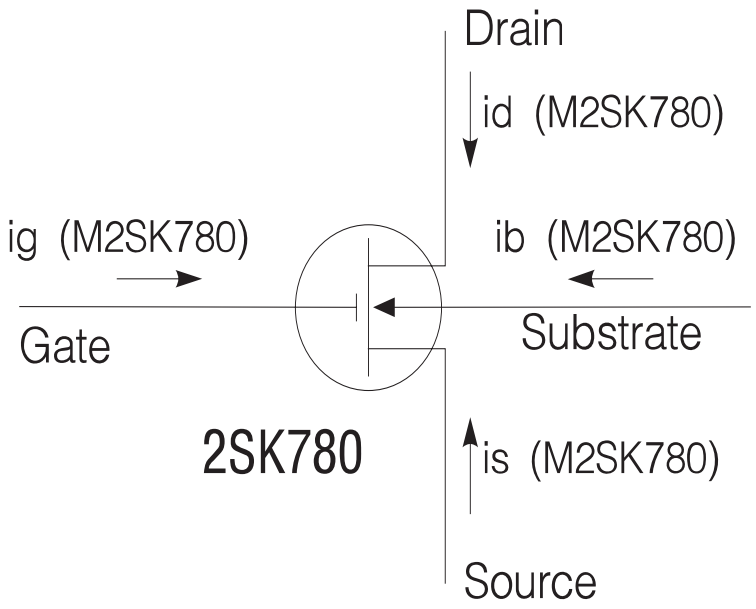

Figure 10-8: Channel MOSFET Current Convention (p-channel currents are opposite).

MOSFET Equivalent Circuits

MIXEDMODE uses two equivalent circuits in MOSFET analysis: DC and transient. The components of these circuits form the foundation for all element and model equations. The DC equivalent circuit is the same as that of transient analysis, except that capacitances are not included.

The basic component in the equivalent circuit is the drain-to-source current ids. The ids equation accounts for all DC currents of the MOSFET model. The gate capacitance, and source and drain diodes are considered separately from the ids equation.

A MOS transistor is described using both the element and the .MODEL statements. The element statement defines the connectivity of the transistor, as well as referencing the .MODEL statement. The .MODEL statement specifies either an n- or p-channel device, the level of the model, and a number of user-specified model parameters.

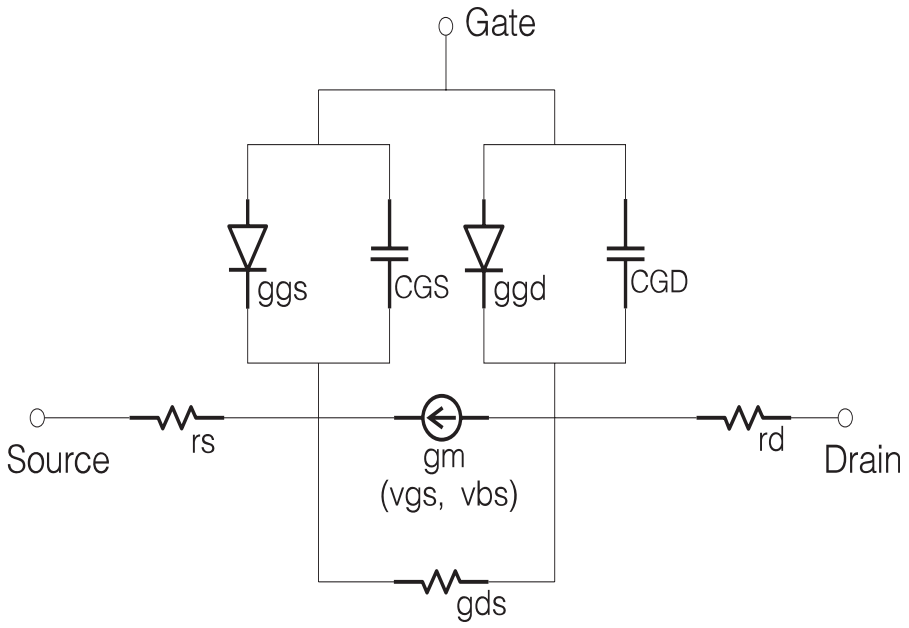

Figure 10-9: MOSFET Equivalent Circuit for Transient Analysis

Table 10-12. MOSFET Model Parameters

Parameters	Description	Units	Default	Area
TOX	Oxide thickness meter m	$1.0*10^{-7}$		
KP	Transconductance parameter	A/V^2	$2.0*10^{-5}$	
NSUB	Substrate doping	$1/cm^3$	0	
GAMMA	Bulk threshold parameter	$v^{0.5}$	0	
THETA	Mobility modulation	0	0	
KAPPA	Saturation field factor	0.2		
PHI	Surface potential	v	0.6	
DELTA	Width effect on threshold voltage	0		
ETA	Static feedback	0		
VTO	Zero-bias threshold voltage	v	0	
RD	Drain ohmic resistance	ohm	0	
RS	Source ohmic resistance	ohm	0	
RSH	Drain and source diffusion sheet resistance	ohm/sq	0	

Table 10-12. MOSFET Model Parameters				
Parameters	**Description**	**Units**	**Default**	**Area**
CGDO	Gate-drain overlap capacitance per meter channel width	F/m	0	
CSDO	Gate-source overlap capacitance per meter channel width	F/m	0	
CGBO	Gate-source overlap capacitance per meter channel width	F/m	0	
CJ	Zero-bias bulk junction bottom cap. per meter of junction area	F/m^2	0	
PB	Bulk junction potential	v	0.8	
MJ	Bulk junction bottom grading coef.		0.5	
CJSW	Zero-bias bulk junction sidewall cap. per meter of junction perimeter	F/m	0	
MJSW	Bulk junction sidewall grading coefficient		0.33	
FC	Coefficient for forward-bias depletion capacitance formula		0.5	
CBD	Zero-bias B-D junction capacitance	F	0	
CBS	Zero-bias B-S junction capacitance	F	0	
JS	Bulk junction saturation current per sq. meter of junction area	A/m^2	$1.0*10^{-8}$	
IS	Bulk junction saturation current	A	$1.0*10^{-14}$	
NSS	Surface state density	$1/cm^2$	0	
NFS	Fast surface state density	$1/cm^2$	0	
TPG	Type of gate material: +1 opposite to substrate -1 same as substrate 0 Al gate		1.0	
LD	Lateral diffusion	m	0	
U0	Surface mobility	$cm^2/v\text{-}s$	600	
TNOM	Parameter measurement temperature	^oC	27	
VMAX	Maximum drift velocity of carriers	m/s	0	
XD	Depletion layer width	m	0	
XJ	Metallurgical junction depth	m	0	

DC Current Equations

The DC current equations describe how the model calculates the drain current ids. In these equations:

vgs is the gate intrinsic source voltage.

vgd is the gate intrinsic drain voltage.

vds is the intrinsic drain-intrinsic source voltage.

vbs is the bulk intrinsic source voltage.

vbd is the bulk- intrinsic drain voltage.

vth is the thermal potential.

Cutoff Region

ids = 0 for vds ≤ vth 10-169

On Region

$$ids = BETA \cdot \left(v_{gs} - vth - \frac{1+fb}{2} \cdot vde \right) \cdot vde \qquad \text{for } v_{gs} > vth \qquad 10\text{-}170$$

where:

$$BETA = KP \cdot \frac{w_{eff}}{l_{eff}} \qquad 10\text{-}171$$

$$= u_{eff} \cdot COX \cdot \frac{w_{eff}}{l_{eff}} \qquad 10\text{-}172$$

vde = min (vds, vdsat) 10-173

and:

$$fb = fn + \frac{GAMMA \cdot fs}{4 \cdot (PHI + vsb)^{1/2}} \qquad 10\text{-}174$$

The narrow width effect is included though the fn parameter:

$$fn = \frac{\varepsilon si \cdot \pi \cdot DELTA}{2 \cdot COX \cdot w_{eff}} \qquad 10\text{-}175$$

The term `fs` expresses the effect of the short channel, and is given by:

$$fs = 1 - \frac{XJ_{scaled}}{l_{eff}} \qquad 10\text{-}176$$

$$\cdot \left\{ \frac{LD_{scaled} + wc}{XJ_{scaled}} \cdot \left[1 - \left(\frac{wp}{XJ_{scaled} +} \right)^2 \right]^{1/2} - \frac{LD_{scaled}}{XJ_{scaled}} \right\} \qquad 10\text{-}177$$

$$wp = xd \ (PHI + vsb)^{1/2} \quad \text{10-178}$$

$$xd = \left(\frac{2 \cdot \varepsilon si}{q \cdot NSUB}\right)^{1/2} \quad \text{10-179}$$

$$wc = XJ_{scaled} \quad \text{10-180}$$

$$\cdot \left[0.0831353 + 0.08013929 \cdot \frac{wp}{(XJ_{scaled})} - 0.0111077 \cdot \left(\frac{wp}{XJ_{scaled}}\right)^2\right] \quad \text{10-181}$$

Effective Channel Length and Width

$l_{eff} = L_{scaled} + XL_{scaled} - 2 \cdot LD_{scaled}$ 10-182

$w_{eff} = W_{scaled} + XW_{scaled} - 2 \cdot WD_{scaled}$ 10-183

Threshold Voltage

$$vth = vbi - \frac{8.14e \times 10^{-22} \cdot ETA}{COX \cdot l_{eff}^3} \cdot v_{ds} + GAMMA \cdot fs \quad \text{10-184}$$

$$\cdot (PHI + v_{sb}^{1/2}) + fn \cdot (PHI + vcb) \quad \text{10-185}$$

where:

$vbi = vgb + PHI$ 10-186

or:

$vbi = VTO - GAMMA \cdot PHI^{1/2}$ 10-187

Saturation Voltage

MIXEDMODE calculates the saturation voltage due to channel pinch-off at the drain side. VMAX is used to include the reduction of the saturation voltage due to carrier velocity saturation effects.

$$vsat = \frac{v_{gs} - vth}{1 + fb} \quad \text{10-188}$$

$$vdsat = vsat + vc - (vsat^2 + vc^2)^{1/2} \quad \text{10-189}$$

where

$$vc = \frac{VMAX \cdot l_{eff}}{us} \quad \text{10-190}$$

The parameter, us, is surface mobility (see Effective Mobility).vdst = vsat

10-191

Effective Mobility

The carrier mobility reduction due to the normal field is modeled as us, the effective surface mobility:

$$us = 1\frac{U0}{+ THETA \cdot (v_{gs} - vth)} \text{ for } v_{gs} > v_{th} \quad \text{10-192}$$

If VMAX is specified, the degradation of mobility due to the lateral field and carrier velocity saturation is determine as:

$$u_{eff} = \frac{us}{1 + \frac{vde}{vc}} \quad \text{for VMAX} > 0 \quad \text{10-193}$$

else

$$u_{eff} = us \quad \text{10-194}$$

Channel Length Modulation

For vds > vdsat, the channel length modulation factor is calculated. The channel length reduction, Δl, is determined depending on the value of VMAX.

$$\Delta 1 = xd \cdot [KAPPA \cdot (vds - vdsat)]^{1/2} \quad \text{10-195}$$

$$\Delta 1 = -\frac{ep \cdot xd^2}{2} + \left[\left(\frac{\exp \cdot xd^2}{2}\right) + KAPPA \cdot xd^2 \cdot (vds - vdsat)\right]^{1/2} \quad \text{10-196}$$

For VMAX > 0, ep is the electric field at the pinch-off point. Its value is approximated by:

$$ep = \frac{vc \cdot (vc + vdsat)}{l_{eff} \cdot vdsat} \quad \text{10-197}$$

The current ids in saturation region is computed as:

$$ids = \frac{ids}{1 - \frac{\Delta 1}{l_{eff}}} \quad \text{10-198}$$

To prevent the denominator from going to zero, the value of Δl is limited by:

$$\Delta 1 = l_{eff} - \frac{(l_{eff}/2)^2}{\Delta 1} \text{ for } \Delta 1 > l_{eff}/2 \quad \text{10-199}$$

Subthreshold Current

The subthreshold current region of operation is characterized by the fast surface state parameter, NFS. The modified threshold voltage, von, is determined by:

von = vth + fastfor NFS >0

where:

$$= v_t \cdot \left[1 + \frac{q \cdot NFS}{COX} + \frac{GAMMA \cdot fs \cdot (PHI + vsb)^{1/2} + fn \cdot (PHI + vsb)}{2 \cdot (PHI + vsb)}\right] \quad 10\text{-}200$$

The current ids is given by:

ids =ids (v_{on}, v_{de}, v_{sb}) · e $^{(v\ sub\ gs\text{-}von)/fast}$ for $v_{gs} < v_{on}$ 10-201

ids = ids (v_{gs}, v_{de}, v_{sb}) 10-202

Capacitances

For CBS = 0 and CBD = 0: 10-203

Cbs = AS · CJ · Cbsj + PS · CJSW · Cbss 10-204

Cbd = AD · CJ · Cbdj + PD · CJSW · Cbds 10-205

Otherwise:

Cbs = CBS · Cbsj + PS · CJSW · Cbss 10-206

Cbd = CBD · Cbdj + PD · CJSW · Cbds 10-207

For Vbs ≤ FC · PB:

Cbsj = (1 – FC)$^{-(1+MJ)}$ · (1 - FC · (1 + MJ) + MJ · Vbs/PB) 10-208

Cbds = (1 – FC)$^{-MJSW}$ 10-209

For Vbd ≤ FC · PB

Cbdj = (1 – Vbd/PB)$^{-MJ}$ 10-210

Cbds = (1 – Vbd/PBSW)$^{-MJSW}$ 10-211

For Vbd > FC · PB:

Cbsj = (1- FC)$^{-(1+MJ)}$ · (1 - FC · (1 + MJ) + MJ · Vbd/PB) 10-212

Cbds = (1 – FC)$^{-MJSW}$

(1 - FC · (1 + MJSW) + MJSW · Vbd/PB) 10-213

Gate Overlap Capacitances

Cgs =CGSO · W 10-214

Cgd =CGDO · W 10-215

Cgb =CGBO · L 10-216

Gate Capacitance Calculation

Define:

cap = COX_{scaled} · w_{eff} · l_{eff} 10-217

Gate-Bulk Capacitance (cgb)

Accumulation: cgb = cap for vgs ≤ von – PHI (10-218)

Depletion: cgb = cap $\frac{von - v_{gs}}{PHI}$ for vgs < von (10-219)

Strong inversion: cgb = 0 for vgs ≥ von (10-220)

Gate-Source Capacitance (cgs)

Accumulation:

cgs = 0 for vgs ≤ von – PHI/2 (10-221)

Depletion:

$$cgs = CF5 \cdot cap \cdot \left(\frac{v_{gs} - vth}{0.5 \cdot PHI}\right) + 1 \quad \text{for vgd} \le \text{von} \tag{10-222}$$

Strong inversion saturation region:

cgs = 2/3 · cap for vgs > von, and vds ≥ vdsat (10-223)

Strong inversion linear region:

For vgs > von, and vds < vdsat

$$cgs = 2/3 \cdot cap \cdot \left\{1 - \left[\frac{vdsat - vds}{2 \cdot vdsat}\right]^2\right\} \tag{10-224}$$

Gate-Drain Capacitance (cgd)

The gate-drain capacitance has value only in the linear region.

Strong inversion linear region:

For vgs > von and vds < vdsat

$$cgd = 2/3 \cdot cap \cdot \left\{1 - \left[\frac{vdsat - vds}{2 \cdot vdsat - vds}\right]^2\right\} 1 \tag{10-225}$$

User-Defined Two-Terminal Elements

Overview

MIXEDMODE users who have acquired the C interpreter can define their own two-terminal elements using the B statement and a function written in C that defines the behavior of the element.

User-Defined Model

A user-defined model is specified by defining:

The dependencies of the device terminal current on the terminal voltages

The derivatives of the terminal current with respect to the terminal voltages

The device current (I) is described by the following equation:

$$I = F1(U,t) + F2(U,t) \cdot \left(\frac{dU}{dt}\right) \quad \text{10-226}$$

where:

U is the device voltage, t is time, and F1 and F2 are functions that determine the behavior of the device. The first term on the right hand side describes the “DC” current and the second term describes “capacitive” current.

The following four functions are specified by the user:

1. F1(U,t) - the “DC” current.
2. F2(U,t) - the “capacitance.”
3. dF1(U,t)/dU - the “DC” differential conductance.
4. Q - the charge associated with F2(U,t).

To define the element, the user prepares a text file that contains an appropriate function written in C. A template for this user-defined function is shown below:

```
int udef(v,temp,ktq,time,curr,didv,cap,charge)
double v;
double ktq;
double time;
double *curr;
double *didv;
double *cap;
double *charge;
{
/* user-supplied code here  */
return(0);
}
```

Input Parameters

Four input parameters are supplied to the function and can be used in the user-defined code. The input parameters are:

v	the voltage across the element (V)
temp	the temperature (K)
ktq	the thermal voltage kT/q (V)
time	transient time (sec); a value of 0 is supplied during DC calculations

Output Parameters

The four output parameters that must be returned by the function are:

curr	the value of F1 (Amps)
didv	the value of dF1(v, time)/dU (A/V)
cap	the value of F2(v, time)
charge	the value of the charge (Q)

Example

Consider an element that consists of a resistor R and a capacitor C connected in parallel. The equation for the total current through this combination is:

$$I(U,t) = \frac{U}{R} + C \cdot \left(\frac{dU}{dt}\right) \quad \text{10-227}$$

The quantities that must be defined by the user are:

$$F1(U,t) = \frac{U}{R} \quad \text{10-228}$$

$$F2(U,t) = C \quad \text{10-229}$$

$$\frac{dF(U,t)}{dU} = \frac{1}{R} \quad \text{10-230}$$

$$Q = C \cdot U \quad \text{10-231}$$

When R=2kΩ and C=100pF, a user-defined function could have the following form:

intrc(v,temp,ktq,time,curr,didv,cap,charge)

```
double v;
double temp;
double ktq;
double time;
double *curr;
double *didv;
double *cap;
double *charge;
{
     *(curr) = v/2000.0;
     *(didv) = 1.0/2000.0
     *(cap) = 1.0e-10;
     *charge)=1.0e-10*v;
     return(0);                    /* 0 - ok */
}
```